

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедри обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТИПЕНКО

«__»_____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

на тему: «Система аналізу даних студентів»

Виконав:

студент IV курсу, групи ІО-61

Валько Антон Вікторович _____

Керівник:

Асистент

Регіда Павло Геннадійович _____

Консультант з нормоконтролю:

Професор кафедри ОТ, д.т.н.

Сімоненко Валерій Павлович _____

Рецензент:

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.467800.003 ПЗ	Пояснювальна записка	70	
3	A4	ІАЛЦ.467800.003 ОА	Опис альбому	1	
4	A4	ІАЛЦ.467800.003 ТЗ	Технічне завдання	1	
5	A4	ІАЛЦ.467800.004 Д1	Додаток 1	1	
6	A4	ІАЛЦ.467800.005 Д2	Додаток 2	1	
7	A4	ІАЛЦ.467800.006 Д3	Додаток 3	1	
8	A4	ІАЛЦ.467800.007 Д4	Додаток 4	13	

				ІАЛЦ.467800.003		
	ПІБ	Підп	Дата			
Розробн.	Валько А. В.			Відомість дипломного проєкту	Лист	Листів
Керівн.	Регіда П. Г.				1	1
Консульт.					КПІ ім. Ігоря Сікорського Каф. ОТ Гр. ІО-61	
Н/контр.	Сімоненко В. П.					
Зав.каф.	Стіренко С. Г.					

**Пояснювальна записка
до дипломного проєкту
на тему: «Система аналізу даних студентів»**

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сергій СТИРЕНКО

«___» _____ 2020 р.

ЗАВДАННЯ
на дипломний проєкт студенту
Валько Антону Вікторовичу

1. Тема проєкту «Система аналізу даних студентів», керівник проєкту асистент Регіда Павло Геннадійович, затверджені наказом по університету від «07» травня 2020 р. №1081-с.
2. Термін подання студентом проєкту: 15.06.2020.
3. Вихідні дані до проєкту: технічна документація. теоретичні та статистичні дані.
4. Зміст пояснювальної записки; аналіз та огляд рейтингової системи оцінювання, побудова та реалізація моделі даних інформаційно-аналітичної системи, реалізація візуально-аналітичного модуля.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): 41 рисунок, 3 блок-схеми.
6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сімоненко В. П. професор кафедри ОТ, д.т.н.		

7. Дата видачі завдання: 09.11.2019.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Затвердження теми роботи	01.09.2020	
2	Вивчення та аналіз завдання	10.11.2019	
3	Розробка архітектури та загальної структури систем	17.01.2020	
4	Розробка структур окремих підсистем	07.02.2020	
5	Програмна реалізація системи	05.03.2020	
6	Оформлення пояснювальної записки	10.05.2020	
7	Захист програмного продукту	15.05.2020	
8	Передзахист	31.05.2020	
9	Захист	19.06.2020	

Студент

Антон ВАЛЬКО

Керівник

Павло РЕГІДА

АНОТАЦІЯ

В бакалаврській дипломній роботі реалізовано систему аналізу даних студентів. Дана інформаційно-аналітична система має можливість зберігати, аналізувати та графічно відображати статистичні дані. Система дає змогу зрозуміти успішність студентів, ефективність предметів та коректність навчального процесу.

Для реалізації системи було використано аналітичну платформу SAP Analytics Cloud та середовище розробки SAP Analytics Cloud Analytics Designer. Даний інструмент разом з мовою JavaScript дозволив налаштувати гнучку поведінку та взаємодію елементів візуалізації. Також було використано мову програмування R для побудови специфічних діаграм.

Система може використовуватись студентами для моніторингу своєї успішності, викладачами для загального аналізу ефективності навчального процесу та керівництвом кафедри для збору статистичних даних.

ANNOTATION

In this work for a Bachelor's Degree the students data analysis system is implemented. This information-analytical system is able to store, analyze and graphically display relevant data. The system makes it possible to understand the students success, the effectiveness of the subjects and the correctness of the educational process.

The SAP Analytics Cloud platform and the SAP Analytics Cloud Analytics Designer were used to implement the system. This toolkit, along with JavaScript, has made it possible to customize the flexible behavior and interaction of the visualization elements. To build specific diagrams was used programming language R. Students can use this system to monitor their performance, faculty can use it to analyze the overall effectiveness of the learning process and department heads can use it to collect statistics.

ЗМІСТ

ЗМІСТ	1
РОЗДІЛ 1 ОГЛЯД ТЕОРЕТИЧНИХ ВІДОМОСТЕЙ.....	5
1.1. Огляд діючої рейтингової системи оцінювання	5
1.2. Основні відомості про бази даних.....	8
1.3. Аналіз конкурентів.....	11
1.4. Аналіз існуючих хмарних систем для бізнес-аналізу	13
Висновки до розділу 1	16
РОЗДІЛ 2 АНАЛІЗ ТА ОГЛЯД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	17
2.1. Опис вибору системи	17
2.2. Огляд програмного забезпечення SAP	18
2.3. Огляд платформи SAP Analytics Cloud.....	19
2.4. Особливості інструменту SAP Analytics Cloud Analytics Designer.....	23
2.5. Опис вхідних даних.....	24
Висновки до розділу 2.....	27
РОЗДІЛ 3 ПРОЕКТУВАННЯ АНАЛІТИЧНОЇ СИСТЕМИ.....	28
3.1. Створення ознак та величин.....	28
3.2. Створення моделі даних	31
3.3. Завантаження даних студентів в систему	34
3.4. Побудова аналітичного додатку	36
Висновки до розділу 3.....	53
РОЗДІЛ 4 ВІЗУАЛІЗАЦІЯ ДАНИХ АНАЛІТИЧНОЇ СИСТЕМИ	54

					ІАЛЦ.467800.003 ПЗ					
					Система аналізу даних студентів					
Зм.	Аркуш	№ Докум.	Підпис	Дата						
	Розробив	Валько А. В.								
	Перевірів	Регіда П. Г.								
	Т. Контр.				КПП ім. Сікорського ФІОТ Група ІО-61					
	Н. контр.	Сімоненко В.П.								
	Затвердив	Регіда П. Г.								
					Аркуш 1 Аркушів 70					

4.1. Сторінка аналітичного додатку «Загальний огляд»	54
4.2. Сторінка аналітичного додатку «Аналіз оцінок»	56
4.3. Сторінка аналітичного додатку «Детальний аналіз»	60
4.4. Виринаюче вікно аналітичного додатку «Аналіз студента»	61
4.5. Виринаюче вікно аналітичного додатку «Аналіз предмету»	64
Висновки до розділу 4.....	67
ЗАГАЛЬНІ ВИСНОВКИ	68
СПИСОК ЛІТЕРАТУРИ.....	69

ВСТУП

Університети зазвичай надають рекомендації щодо застосування рейтингової системи оцінювання результатів навчання. Вони призначені для викладачів та користувачів рейтингових систем оцінювання, а також для співробітників деканатів. В рекомендаціях розкриваються принципи побудови та функціонування рейтингових систем оцінювання результатів навчання студентів. Однак недотримання та порушення даних рекомендації відслідкувати досить важко. Зазвичай цим ніхто не займається, оскільки це потребує значних трудовитрат.

В університеті прийнято використовувати універсальну стобальну шкали оцінювання. Це дозволяє зробити результат оцінювання навчальних досягнень студентів набагато більш порівнянними, а застосування цієї системи при вирішенні питань стипендіального забезпечення студентів може зробити та робить цю процедуру набагато більш прозорою і обґрунтованою.

Однак інформацію щодо своєї успішності студенту зазвичай дізнатись проблемно, або інколи навіть неможливо. Оскільки немає єдиної системи, яка б об'єднувала усі дані та оцінки студентів, з'являється потреба у створенні такої системи.

Звичайно, для контролю та оцінювання студентів існують залікові книжки, проте вони в своєму традиційному вигляді давно застаріли. На сьогоднішній день інформаційні технології розвиваються дуже стрімко, і така тенденція дає змогу для зберігання великих масивів даних, швидкої обробки та аналізу цих даних. Також з'являється можливість автоматизації процесів, які раніше виконувалися вручну. Саме тому для підвищення ефективності навчального процесу та ліквідації зайвого документообігу було розроблено систему аналізу даних студентів, яка являє собою залікову книжку в режимі «онлайн» з гнучким функціоналом аналізу даних. Система надаватиме достовірну та прозору інформацію.

					ІАЛЦ.467800.003 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

Однією з переваг розробленої інформаційно-аналітичної системи є її розміщення в «хмарі». Тобто на сервісі, який відкидає потребу у зберіганні даних локально на комп'ютері чи телефоні. Дана платформа дозволяє отримувати доступ до потрібних даних миттєво з будь-якого пристрою, який має підключення до мережі інтернет.

Змінювати та завантажувати нові дані в систему зможуть лише викладачі. Студенти матимуть змогу переглядати оцінки, аналізувати свою ефективність та досягнення. Деканат зможе приймати управлінські рішення на основі статистичних даних та корисних підказок, які система створюватиме автоматично, а також матиме можливість контролювати діяльність студентів і викладачів.

Система дасть змогу відслідковувати коректність застосування рейтингових систем оцінювання студентів по кожному предмету. Такий підхід змусить викладачів ретельніше підходити до оцінювання студентів для уникнення непорозумінь. Студенти, в свою чергу, отримають систему з релевантною інформацією по оцінюванню.

Саме тому метою бакалаврської роботи є реалізація системи аналізу даних студентів. Завданнями роботи є побудова архітектури та реалізація моделі інформаційно-аналітичної системи, створення додатку з візуалізаціями, проведення аналізу застосування рейтингової системи оцінювання студентів.

Об'єктом завдання є прототип інформаційно-аналітичної системи аналізу даних студентів. Система повинна зберігати завантажені в неї дані та аналітики, аналізувати їх та будувати візуалізації на їх основі.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

РОЗДІЛ 1 ОГЛЯД ТЕОРЕТИЧНИХ ВІДОМОСТЕЙ

1.1. Огляд діючої рейтингової системи оцінювання

В основу рейтингових систем оцінювання (РСО) результатів навчання в студентів в КПІ з кожного кредитного модуля закладено контроль і накопичення рейтингових балів за різноманітну діяльність студентів протягом всього процесу навчання [1]. В університеті прийнято використовувати універсальну сто-бальну шкали оцінювання. Це дозволяє зробити результат оцінювання навчаль-них досягнень студентів набагато більш порівнянними, а застосування цієї сис-теми при вирішенні питань стипендіального забезпечення студентів може зро-бити та робить цю процедуру набагато більш прозорою і більш обґрунтованою. Такий підхід явно переважає за змістом за будь-які інші відомі форми оціню-вання.

Метою рейтингової системи оцінювання є:

- покращення та посилення навчального процесу, а також підвищення якості підготовки студентів;
- збільшення мотивації студентів до активного навчання та система-тичної самостійної роботи протягом навчального семестру та розу-міння відповідальності за результати навчальної діяльності протя-гом семестру;
- встановлення постійного зворотного зв'язку з кожним студентом та викладачем, шляхом своєчасного коригування його навчальної дія-льності;
- забезпечення та утримання правильної конкуренції у навчанні;
- збільшення об'єктивності оцінювання результатів навчання;

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

- зменшення емоційних, психологічних і фізичних навантажень у періоди екзаменаційних сесій та екзаменів.

Опис рейтингової системи оцінювання є складовою робочої програми кредитного модуля або ж додатком до неї. РСО також є складовою рекомендацій до засвоєння студентами будь-якого кредитного модуля.

Рейтингова система оцінювання результатів навчання студентів повинна бути стандартизованою та чітко формалізованою за своєю структурою. Для цього, аналізуючи та виходячи зі значення вагових балів та оцінок, розробляються критерії оцінювання» для кожного контрольного заходу з визначенням певних рівнів засвоєння навчального матеріалу та сформованості вмінь.

Рейтингова оцінка трансформується до спеціальної університетської системи оцінювання згідно з таблицею 1.1.

Таблиця 1.1

Рейтингові бали	Оцінка за університетською шкалою
95 ... 100	Відмінно
85 ... 94	Дуже добре
75 ... 84	Добре
65 ... 74	Задовільно
60 ... 64	Достатньо
Менше 60	Незадовільно

Як правило, рейтинговий лист ведеться лектором або під керівництвом лектора викладачем, який проводить заплановані практичні або лабораторні заняття у навчальній групі. Рейтинговий лист ведеться у довільній формі, проте він має містити список студентів навчальної групи, перелік контрольних заходів (точок контролю), балів, що отримав кожен студент на контрольних заходах, а

також кількість підсумкових балів наприкінці семестру. Доцільним є ведення рейтингового листа в онлайн вигляді. Ще однією із вимог є те, що рейтинг-лист повинен зберігатися протягом навчального року.

Оцінки кожного контрольного заходу (тобто контрольних робіт, лекцій та різних практичних занять) мають виконувати детальну розпізнавальну та певну мотиваційну функції. Саме із цих вимог випливає та формується рекомендація викладачам щодо застосування при оцінюванні студентів навчальної групи або ж потоку всієї шкали оцінок. Ідеальний розподіл позитивних оцінок поточного контролю значної групи студентів має нагадувати форму дзвону з медіаною по центру (рис. 1.1).

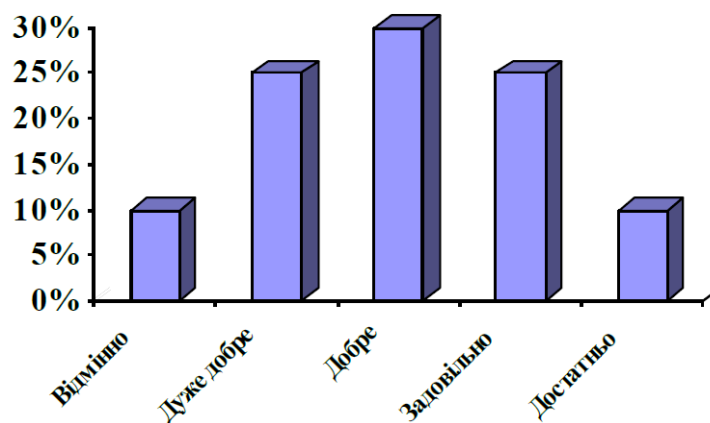


Рис. 1.1. Рекомендований розподіл оцінок

Зміщення рекомендованого розподілу оцінок в будь-який бік є проявом аномалій організації та проведення контролю або навчання. Як правило, будь-яке зміщення рекомендованого розподілу праворуч, тобто занадто велика частка оцінок «відмінно», свідчить про низький рівень запланованих та проведених результатів навчання (рівень навчання) або недостатній рівень формальної складності контрольних завдань. Також це може бути наслідком низьких критеріїв

оцінювання або зниженої вимогливості викладача. Зміщення розподілу ліворуч, тобто велика частка незадовільних результатів є результатом слабкої умотивованості студентів або їхньої низької попередньої підготовки. Це також може бути наслідком недостатнього інформаційно-методичного забезпечення навчальної дисципліни. При наявності значної деформації оцінок з певного предмету кафедра, тобто її завідувачі, муть є провести аналіз різних можливих причин та провести необхідні відповідні педагогічні заходи. На жаль, такі випадки деформацій не є рідкістю, а отже на це треба звертати додаткову увагу, аби запобігати некоректного рівня навчання або невідповідності навчального процесу зі стандартизованими нормами.

1.2. Основні відомості про бази даних

База даних – це сукупність даних, яка зберігає певну організовану інформацію. В основному база даних містить наступні елементи: таблиці, схеми, подання або представлення та інші допоміжні об'єкти [2]. Кожна з таблиць у базі даних містить поля, які стосуються інформації, що зберігається в таблиці. Сучасна база даних, окрім лише даних, може містити їх опис та різноманітні засоби для їх обробки. Проте в загальному випадку під базою даних можна розуміти будь-який впорядкований набір інформації та будь-яких даних.

Разом із базами даних обов'язково розглядається поняття СКБД – система керування базами даних. СКБД – це певний комплекс програмних засобів, який потрібен для створення структури нової бази даних, або її наповнення, контролювання, керування та редагування вмісту або ж відображення необхідної інформації.

СКБД є основою створення програм користувача для будь-яких предметних областей. На вибір СКБД впливають безліч факторів, серед яких наступні:

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

наявне програмне та технічне забезпечення, вимоги до продуктивності при обробці даних, тип моделі даних, що підтримується, оперативна та дискова пам'ять, потреби функціональних програм користувача, що розробляються, кваліфікація користувачів та інші фактори. Після вибору та встановлення СКБД створюється база даних, де задається її структура, здійснюється ведення та керування даними, а також виконуються будь-які дії, що передбачені функціональними можливостями встановленої бази даних.

Існує велика кількість різновидів баз даних, що відрізняються за різними критеріями [3]. За моделлю організації даних розрізняють наступні бази даних:

- Ієрархічні;
- Мережні;
- Реляційні;
- Об'єктні;
- Об'єктно-орієнтовані;
- Об'єктно-реляційні.

За способом доступу майже всі даних бази даних поділяються на бази даних, які мають локальний доступ і бази даних з дистанційним або так званим мережевим доступом [4]. Системи централізованих баз даних з мережевим доступом припускають різноманітні архітектури подібних систем.

MySQL – являє собою вільну, тобто безплатну систему управління базами даних. Цю систему управління базами даних з повністю відкритим кодом було створено як альтернативу комерційним рішенням. Дана база даних використовується для створення різноманітних динамічних веб-сторінок, оскільки має гарну підтримку з боку різноманітних новітніх мов програмування. Зазвичай база даних MySQL використовується як певний сервер, до якого звертаються локальні або дистанційні клієнти, проте до дистрибутиву входить

бібліотека внутрішнього сервера, що дозволяє включати базу даних MySQL до класу автономних програм. Гнучкість СКБД MySQL з легкістю забезпечується підтримкою великої кількості типів таблиць: користувачі БД можуть вибрати як таблиці, що підтримують повнотекстовий пошук, так і таблиці, що підтримують транзакції на рівні окремих записів.

PostgreSQL - об'єктно-реляційна система управління базами даних. Порівняно до інших проектів з відкритим кодом, PostgreSQL не контролюється якоюсь однією компанією, а її розробка можлива лише завдяки одночасній роботі багатьох груп людей та компаній, які бажають використовувати цю СКБД та впроваджувати в неї самі нові розробки, технології та досягнення. Тобто код даної бази даних можна спокійно знайти в інтернеті та вдосконалити власноруч за умови наявності потрібних навичок.

СКБД Oracle – це програмний комплекс та програмне забезпечення, що дозволяє створювати додатки будь-якої складності. Основою цього комплексу є база даних, що зберігає інформацію, кількість якої за рахунок наданих засобів масштабування можна сказати що практично необмежена. З високою ефективністю працювати та виконувати операції з цією інформацією одночасно може практично будь-яка кількість користувачів, не проявляючи тенденції до зменшення та зниження продуктивності системи при різкому та стрімкому збільшенні їхньої кількості. Механізми масштабування в системі керування базами даних Oracle останніх версій дозволяють необмежено збільшувати потужність і швидкість роботи сервера Oracle і своїх додатків, просто додаючи все нові вузли кластеру та комплексу.

Microsoft SQL Server – це система управління реляційними базами даних, яка розроблена компанією Microsoft. Для роботи з нею використовується спеціальна мова запитів - Transact-SQL. В даній СКБД сервер виконує саму головну

функцію по збереженню та наданню даних у відповідь на запити інших застосунків та програмного забезпечення, яке може виконуватися як на тому ж самому сервері, так і у мережі. Microsoft SQL Server використовується як для малих та середніх за розміром баз даних, так і для дуже великих баз даних з масштабом повноцінного підприємства.

Microsoft Office Access – реляційна СКБД, розроблена компанією Microsoft. Дана база даних має дуже широкий спектр функцій, який включає зв'язані запити, зв'язок із зовнішніми таблицями і різноманітними базами даних. Завдяки вбудованій мові VBA (Visual Basic for Application), в самому Access є можливість писати застосунки, що працюють з різними базами даних. До основних компонентів MS Access відноситься будівник таблиць, будівник екранних форм, будівник SQL-запитів, будівник звітів. Оскільки дана СКБД має можливість викликати скрипти мовою VBA, то стає можливою розробка програми і бази даних практично з порожнього листа або написання оболонки для зовнішньої БД. MS Access є файл-серверною СКБД і тому застосовується не для великих масштабів, а лише до маленьких додатків. Також слід додати, що в даній СКБД відсутній ряд механізмів, який є досить необхідним у багатокористувацьких БД.

1.3. Аналіз конкурентів

Метою виконання роботи є реалізація інформаційно-аналітичної системи, яка дозволить завантажувати в неї дані та оцінки студентів, зберігати їх, та аналізувати інструментами візуалізації. В системі буде приводиться як загальна статистика так і детальний огляд студентів і предметів.

Одним із конкурентів даного проекту є система «Електронний Кампус КПП» [5], який надає можливість користуватися накопиченими електронними

					ІАЛЦ.467800.003 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

інформаційними ресурсами для підтримки навчального процесу відповідно до навчальних планів підготовки бакалаврів, спеціалістів та магістрів. Доступ до методичних матеріалів здійснюється за отриманими логінами та паролями, що забезпечує розмежування доступу до інформації. Гранулярність даних, а саме оцінок студентів, в даному рішенні надзвичайно висока: оцінки виставляються з максимальною деталізацією, тобто за кожне завдання, лабораторну роботу та лекцію. Дана система спеціалізується на поточному контролі і є альтернативою паперового варіанту списку студентів із їх оцінками. Однак поточна реалізація системи «Електронний Кампус КП» унеможливило гнучкий аналіз студентом своєї успішності. Неможливим є також перегляд усіх агрегованих балів, аналіз динаміки оцінювання, порівняння з іншими студентами та аналіз коректності застосування рейтингової системи оцінювання по будь-якому предмету. Іншими словами, система не має жодного інструменту для будь-якого аналізу даних.

«НТУУ ім. Ігоря Сікорського» не пропонує інших альтернатив та рішень. Інші університети Києва та України також не можуть запропонувати цікавих та значних інформаційно-аналітичних систем, які можна було би використовувати студентами, викладачами для перегляду та керування своїми даними та оцінками. В мережі інтернет важко знайти інші рішення та системи, які могли б скласти конкуренції системі, яка розробляється в ході виконання дипломного проекту. Пропонуються до уваги збірки із декількох дашбордів та інформаційних панелей, які вже давно втратили свою актуальність та не можуть називатись повноцінними системи для аналізу даних. Дані розроблені рішення можуть слугувати лише певним візуальним представленням даних на вибраний момент часу без гнучкого аналізу та всебічного огляду потрібних даних та без відкритого коду для подальшого налаштування системи.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

1.4. Аналіз існуючих хмарних систем для бізнес-аналізу

Програмне забезпечення для бізнес-аналітики - це сукупність інструментів, які дозволяють створювати цінну аналітичну інформацію із великих об'ємів даних. Технології бізнес-аналізу включають сховища даних, інформаційні панелі, спеціальні звіти, засоби пошуку даних та хмарні дата-сервіси [6]. Інструменти бізнес-аналітики мають величезне значення для якісного вимірювання основних показників ефективності на всіх рівнях організації. Такі інструменти дозволяють компаніям бачити як історичні, так і поточні дані в контексті, що дозволяє краще приймати рішення та прогнозувати розвиток залежно від цілей бізнесу.

Бізнес-аналітика полягає у наданні потрібної інформації у потрібний час, а хмарні інструменти забезпечують легкий та швидкий доступ до ВІ-додатків. Перевага додатків Cloud BI полягає в тому, що вони доступні на багатьох пристроях та веб-браузерах [7].

Хмарне програмне забезпечення для бізнес-аналізу дозволяє використовувати інструменти для бізнес-аналізу без потреби у встановленні, оновленні та зберіганні локально на комп'ютері [8]. Це дозволяє користувачам концентруватися на розробці звітів та не витратити час на вирішення технічних проблем. Саме тому хмарні технології та бізнес-аналітика ідеально доповнюють один одного.

Пошук найкращої ВІ-платформи для конкретних цілей є складним завданням і потребує глибокого аналізу, оскільки на ринку існує безліч конкурентоспроможних продуктів [9], які пропонують різноманітні можливості для вирішення проблем бізнесу та продаються за різними цінами.

Найпопулярніші хмарні програмні забезпечення для бізнес-аналізу [10]:

- Microsoft Power BI Pro;

- SAP Analytics Cloud;
- Tableau Online;
- Oracle Analytics Cloud;
- IBM Cognos Analytics.

Усі продукти представлені дуже відомими компаніями [11]. Вони займаються різними видами діяльності, проте розробили свої хмарні BI продукти в погоні за модою та збільшенні кількості зацікавлених осіб. Виключенням із правила є компанія Tableau, яка з самого початку була створена для розробки програмного забезпечення для бізнес-аналітики.

Microsoft Power BI Pro – найдешевший продукт із приведеної п'ятірки [12]. За 10 доларів на місяць ми отримуємо 10 Гб пам'яті в хмарі та повноцінну платформу для аналізу даних з безліччю можливостей. Як і всі інші компанії, Microsoft пропонують безкоштовну пробну версію свого продукту на 30 днів. Не явною перевагою Power BI Pro є наявність альтернативної десктоп версії, яка надається безкоштовно. Це дозволяє вивчати інструменти бізнес-аналізу, не купуючи хмарну платформу. Через свій скромний ціник Microsoft Power BI Pro – це ідеальне рішення для особистого користування.

IBM Cognos Analytics надає менш привабливий та сучасний інтерфейс. Проте може бути гарним рішенням, якщо потрібно інтегруватись з іншим продуктами від IBM [13]. Дана платформа призначена для аналітики з додатковим функціоналом пошуку та аналізу даних. Такий контекст більше спрямований на технічного користувача, тобто користування продуктом має вищий поріг входження та може бути складним для кінцевого бізнес-користувача. Додатково слід зауважити, що популярність IBM Cognos Analytics досить мала, тому можливі проблеми із знаходженням ресурсів для вивчення цього продукту.

Tableau – є беззаперечним лідер на ринку серед компаній, орієнтованих на розробку програмного забезпечення для бізнес-аналізу [14]. Платформа Tableau Online дотримується стратегії надання програмного забезпечення, яке вимагає мінімального вивчення і дозволяє бізнес-користувачам працювати з даними за допомогою візуального аналізу без допомоги професіональних розробників. Варто відзначити, що Tableau Online – це хмарна версія надзвичайно популярної Tableau Desktop. Хмарна BI-платформа демонструє широку розповсюдженість на великих підприємствах та рідко використовується індивідуальними користувачами [15].

SAP Analytics Cloud – продукт, який зробив феноменальний стрибок популярності завдяки своїм проривним технологіям. Переконливі рейтинги додатково підтверджують його конкурентоспроможність [16]. SAP Analytics Cloud об'єднує в собі функціонал бізнес-аналітики, планування та розширеного аналізу. Додатково слід відзначити підтримку повноцінних програм у хмарі, що дає змогу розробляти спеціальні додатки ідеально підлаштовані під потреби користувачів. Клієнти високо оцінюють продукт за сукупними показниками, що виражає високий рівень задоволеності як продавцем, так і його продуктом. Скромна ціна в \$22 за таку багатофункціональну платформу здається заниженою на фоні конкурентів.

Висновки до розділу 1

Було зібрано та проаналізовано основні відомості про діючу рейтингову систему оцінювання. Розглянуто основні відомості про бази даних та виявлено деякі необхідно для подальшої розробки особливості.

Також проаналізовано конкурентів інформаційно-аналітичної системи для аналізу даних студентів. Виявлено недоліки в існуючій системі та описано переваги системи, яка розробляється. Не виявлено додаткових конкурентів для системи, яка розробляється в ході виконання дипломного проекту. Окремо розглянуто хмарне програмне забезпечення для бізнес-аналізу. Здійснено всебічне порівняння продуктів та виявлено як слабкі так і сильні сторони кожного із них.

					ІАЛЦ.467800.003 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 АНАЛІЗ ТА ОГЛЯД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Опис вибору системи

При підготовці до бакалаврської роботи було проаналізовано велику кількість інформаційних джерел та був зроблений висновок про доцільність використання для даної роботи хмарних сервісів – новинок сучасних технологій, які відкидають потребу у зберіганні даних локально на комп'ютері чи телефоні. Дані платформи дозволяють отримувати доступ до потрібних даних миттєво з будь-якого пристрою, який має підключення до мережі інтернет. Наразі хмарний спосіб зберігання даних є найзручнішим та найнадійнішим.

Один із видів хмарних сервісів – це SaaS (Software as a Service – Програмне забезпечення як Послуга). Така модель дозволяє клієнтам отримувати в своє розпорядження певні програмні продукти за допомогою інтернету [17]. В свою чергу, хмарне програмне забезпечення для бізнес-аналізу дозволяє використовувати інструменти для бізнес-аналізу без потреби у встановленні, оновленні та зберіганні локально на комп'ютері. Це дозволяє користувачам концентруватися на розробці звітів та не витрачати час на вирішення технічних проблем. Саме тому хмарні технології та бізнес-аналітика ідеально доповнюють один одного.

Хоча існує багато конкурентоспроможних продуктів, які пропонують бізнес-аналітику в хмарі, вибір зупинився на платформі SAP Analytics Cloud завдяки його проривним технологіям. Переконливі рейтинги додатково підтверджують його конкурентоспроможність. SAP Analytics Cloud об'єднує в собі функціонал бізнес-аналітики, планування та розширеного аналізу. Додатково слід відзначити підтримку повноцінних програм у хмарі, що дає змогу розробляти спеціальні додатки ідеально підлаштовані під потреби користувачів. Клієнти висо-

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

ко оцінюють продукт за сукупними показниками, що виражає високий рівень задоволеності як продавцем, так і його продуктом.

Основні вимоги, які задовольняє вибрана система:

- Зберігання релевантних даних із заданим рівнем деталізації.
- Багатовимірний та гнучкий аналіз отриманих даних.
- Сумісність із різноманітними web-інтерфейсами.
- Оперативність надання інформації.
- Можливості для розмежування доступу до інформації.
- Наявність широкого вибору інструментів для візуалізації даних.

2.2. Огляд програмного забезпечення SAP

SAP – німецька корпорація, розробник програмного забезпечення та надавач послуг консалтингу, яка виробляє корпоративне програмне забезпечення та забезпечує підтримку програм та програмного забезпечення для компаній та підприємств будь-якого розміру по всьому світу [18]. SAP є одним із найбільших розробників корпоративного програмного забезпечення в Європі і четвертим за величиною розробником програмного забезпечення для корпоративного користування у світі з 2009 року. Компанія SAP завжди знаходиться в епіцентрі всіх кардинальних та рішучих нових змін у світі сучасних технологій. Як провідний розробник корпоративного програмного забезпечення вона допомагає організаціям упоратися з негативною тенденцією ускладнення процесів, створити нові можливості для запровадження інновацій і розвитку, а також випередити конкурентів. Напевно, найвідомішим продуктом компанії SAP являється її програмне забезпечення для планування ресурсів підприємства (SAP ERP).

SAP ERP (SAP Enterprise Resource and Planning) – це програмне забезпечення, яке розробляється для організацій, що належать до різних галузей промисловості, незалежно від їх розміру. Модуль ERP розроблений для підтримки та інтеграції майже всіх функціональних напрямків бізнес-процесу, таких як закупівля товарів та послуг, продаж та розповсюдження, фінанси, облік, людські ресурси, виробництво, планування виробництва, логістика та управління.

SAP BW (SAP Business Warehouse) – продукт, орієнтований на зберігання даних. Він збирає, перетворює та зберігає дані, що генеруються у різних додатках чи модулях, та робить їх доступними за допомогою вбудованих інструментів звітності, бізнес-аналізу та аналітики, а також програмного забезпечення сторонніх виробників. SAP BW – це також платформа розробки, яку програмісти використовують для створення та модифікації сховищ даних, виконання завдань управління даними, генерування звітів та розробки програм аналітики.

SAP BI (SAP Business Intelligence) – це продукт SAP, який фокусується на наданні своїм клієнтам зручної форми представлення даних, які можуть бути корисними для аналізу цілей та прийняття бізнес-рішень. SAP BI охоплює різноманітний набір інструментів, об'єднує та завантажує дані з різних систем, тоді як інструменти звітності, такі як конструктор запитів, дизайнер веб-додатків та аналізатор, в основному використовуються для створення звітів, які відображають дані консолідованих із сховища даних для аналізу цілей. Дані можуть бути представлені у формі звітів і можуть бути відображені у вигляді таблиць, діаграм тощо, щоб ефективніше та простіше приймати бізнес-рішення.

2.3. Огляд платформи SAP Analytics Cloud

SAP Analytics Cloud – це повнофункціональне хмарне рішення, розроблений з метою об'єднати всі необхідні функції для бізнес-аналізу та аналізу даних

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

в одному продукті [19]. Крім звичних функцій обробки та візуалізації даних платформа надає можливості планування і прогнозування. Об'єднання таких надзвичайно потужних інструментів в одне рішення робить надлишковим впровадженням додаткових продуктів (Рис. 2.1).

Основні можливості SAP Analytics Cloud:

- Бізнес-аналітика. Аналітичне рішення дозволяє отримати миттєву відповідь на будь-яке бізнес-питання. Надається можливість підключатися до різних джерел, комбінувати дані з них, візуалізувати результати аналізу великих даних і виконувати необхідні дії безпосередньо з програми відразу ж після прийняття бізнес-рішень.
- Планування. Інструменти для спільної роботи і аналізу даних вбудовані безпосередньо в процеси планування, так що користувачам більше не доведеться перемикатися між додатками. Планування та аналіз стають простішими на будь-якому рівні деталізації за участю будь-якої кількості користувачів.
- Прогнозування. Впроваджена функціональність використання методів машинного виявлення, щоб допомогти кінцевим користувачам в створенні прогнозів, пошуку ключових факторів впливу, виявленні аномальних значень і виконанні аналізу альтернативних сценаріїв.

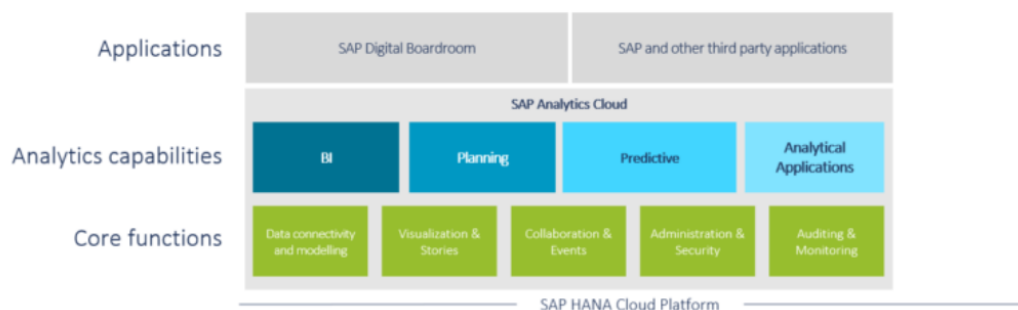


Рис. 2.1 Основні можливості SAP Analytics Cloud

SAP Analytics Cloud дозволяє користувачам продукту працювати з безліччю джерел даних, серед яких SQL бази даних (MySQL, PostgreSQL, Oracle, Microsoft SQL), сервіси Open Data Protocol, Google Диск, Google Таблиці, Google BigQuery, Dropbox, SAP ERP, SAP BW та інші. Також є можливість імпортування файлу з локального комп'ютеру в форматі «.xls» або «.csv».

Моделі – основне поняття зберігання даних в SAP Analytics Cloud. По аналогії з таблицями в звичних базах даних, модель зберігає великий обсяг бізнес-даних з використанням звичайної бізнес-термінології.

В SAP Analytics Cloud існує два типи моделей:

- аналітичні моделі;
- моделі планування.

Аналітичні моделі - це повнофункціональні моделі, які містять ваші бізнес-дані. Зазвичай аналітичну модель використовується, якщо вам потрібно проаналізувати дані, визначаючи тренди і аномалії.

Моделі планування використовуються для того, щоб полегшити виконання завдань планування бізнесу чи будування прогнозу. Працюючи з цим типом моделей, користувачі можуть використовувати широкий набір функцій для оновлення значень в моделі, створення нових значень та користуватись повноцінними функціоналом планування.

Замість того щоб намагатися працювати безпосередньо з величезними обсягами даних, в продукті SAP Analytics Cloud визначаються показники і виміри, що представляють ваші бізнес-дані. Будь-яка модель задається певним набором вимірів і показників, додатково визначаються способи агрегації, масштабування, задаються ієрархії та атрибути.

Показники являють собою кількості, які надають значення даним (наприклад, виручка від продажів, зарплата або число співробітників). Виміри пред-

ставляють собою категорії, які забезпечують сприйняття ваших даних (наприклад, категорія продукту, дата або місце розташування). Разом вони утворюють структуру для перегляду даних зручним способом.

На базі моделі створюється журнали. У процесі роботи з SAP Analytics Cloud вони відіграють центральну роль. Журнали дозволяють використовувати дані для візуалізації інформації за допомогою діаграм і таблиць. Іншими словами, журнал – це дашборд в системі SAP Analytics Cloud.

Однієї із переваг системи SAP Analytics Cloud є використання бази даних HANA – реляційної БД від компанії SAP [20]. База даних підтримує як рядкове, так і стовпчикове зберігання даних: перше використовується для транзакційних навантажувачів, друге - для аналітичних. Індексний сервер забезпечує управління сеансами, авторизацію, управління транзакціями та обробкою команд.

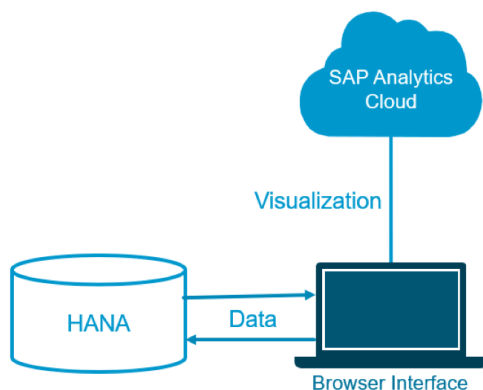


Рис 2.2. Взаємозв'язок БД HANA із SAP Analytics Cloud

База даних HANA знаходиться на спеціальному виділеному сервері, на якому зареєстрована система SAP Analytics Cloud [21]. Таким чином, весь обмін, обробка та зберігання даних відбувається в хмарі, а не на локальному пристрої

користувача, що значно пришвидшує роботу з даними та дуже сильно економить локальну пам'ять пристрою (Рис. 2.2).

2.4. Особливості інструменту SAP Analytics Cloud Analytics Designer

SAP Analytics Cloud Analytics Designer – це розширений функціонал продукту SAP Analytics Cloud [22]. За допомогою дизайнера аналітики можна створювати додатки для аналізу і планування даних. Аналітичні додатки створюються малим числом експертів, але застосовуються багатьма кінцевими користувачами.

Основна відмінність від журналів полягає в тому, що в аналітичних додатках можна налаштувати поведінку графічних елементів за допомогою набору певних подій або скриптів та вказати, які дії повинні відбуватися при ініціації подій [23]. Ця гнучкість дозволяє створювати різні аналітичні програми – від простих статичних інструментальних панелей до складних аналітичних програм з численними опціями для огляду даних і навігації по ним. Розширюючи поняття журналу, аналітичний додаток додає не тільки інтерактивну поведінку графічних елементів, які можна персоналізувати та адаптувати відповідно до вимог користувача, а й нові специфічні об'єкти, які можна створювати та налаштовувати власноруч.

Написання скриптів та подій в аналітичному додатку відбувається за допомогою обмеженого набору мови програмування JavaScript [24] та налаштованих інтерфейсів програмування (API) [25]. В спеціальному середовищі розробки здійснюється написання коду для будь-якого об'єкту програми, щоб впровадити інтерактивність та специфічну поведінку, потрібну для потреб користувача.

Одним із прикладів використання скриптів для вдосконалення роботи додатку є автоматичне заповнення випадального списку елементами виміру вибра-

ної моделі. Наприклад, в аналітичному додатку потрібно здійснювати фільтрацію по містам України. Для цього при його ініціалізації скрипт буде автоматично підчитувати усі елементи виміру «Місто» та додавати його до списку. Також потрібно налаштувати логіку фільтрації, щоб користувач мав змогу вибирати будь-яке значення із заповненого списку, а аналітичний додаток реагував на цей вибір і перебудовувався відповідним чином.

2.5. Опис вхідних даних

Вхідними даними, що були використані під час розробки системи, були оцінки студентів по кожному з предметів, які знаходилися у файлі формату «Excel». Файл містив прізвище, ім'я та по-батькові студентів (більше 120 записів) та повну назву предмету, який викладався (більше 60 записів). Такий зріз даних дав змогу отримати більше 7000 унікальних записів оцінок (рис. 2.3). Слід зазначити, що студенти групувались по групам, що дало змогу отримати додаткову аналітику. В свою чергу, предмети мали 3 додаткових атрибути, які їх описували: семестр, форма контролю та години (рис. 2.4). Все це дало змогу завантажити дані в систему та створити на їх основі модель даних.

1		Алго	Вище	Вступ	Дода	Істор	Комп	Комп	Проф	Фізич	Вище	Диск	Інозе	Комп	Об'єк	Украї	Фізик	Фізич	Вступ	Дода
2	Балащенко Гліб Ігорович	79	60	64	60	70	60	60	60	85	60	60	60	60	74	64	62	94	75	60
3	Валько Антон Вікторович	94	67	60	60	63	85	75	78	90	70	61	65	76	93	88	67	80	80	76
4	Воробйов Анатолій Олекса	90	88	70	77	72	78	65	87	90	67	76	65	76	99	85	70	93	90	65
5	Дем'яненко Максим Олекс	87	75	69	83	70	75	65	82	85	62	65	90	87	95	65	77	95	65	60
6	Добровольський Дмитро Д	91	85	70	95	75	85	88	80	75	75	77	85	89	100	97	82	76	95	85
7	Дудченко Ігор Віталійович	97	95	68	89	95	85	87	97	96	85	95	90	85	100	87	85	94	95	80
8	Іасешвілі Георгій Нугзаров	87	65	61	75	80	60	61	75	60	60	60	67	68	100	80	68	74	90	68
9	Коваленко Антон Сергійов	87	60	70	70	75	63	75	72	75	68	62	65	73	78	72	81	85	80	78
10	Колпак Максим Віталійович	93	65	88	95	95	75	77	85	60	76	67	75	86	100	86	70	60	69	65
11	Косюга Олексій Євгенійови	84	77	80	85	72	92	85	76	95	69	88	100	89	97	98	81	74	95	68
12	Кот Ольга Сергіївна	93	67	61	75	76	78	81	65	85	62	67	95	81	98	77	82	96	93	65
13	Кузьменко Богдан Сергійов	93	65	73	65	75	90	75	73	96	65	60	75	60	95	68	65	94	85	60
14	Левківський Вадим Валерій	93	85	65	92	95	95	96	86	65	75	96	90	100	100	98	85	90	95	85
15	Липай Владислав Сергійов	73	60	60	60	70	60	61	60	60	60	60	60	60	67	65	61	85	65	60
16	Лисенко Дмитро Вадимови	91	78	69	82	60	85	86	99	95	85	96	65	100	98	70	77	95	70	95
17	Літвін Олександр Юрійович	76	60	63	66	60	65	65	64	75	60	60	60	75	73	75	60	60	80	60

Рис. 2.3. Вхідні дані

Предмет	Форма контролю	Години	Семестер
Алгоритми та структури даних	Д.З.	90	1
Вища математика-1. Диференціальне та інтегральне числення	Екз.	195	1
Вступ до операційної системи Linux	зал.	90	1
Додаткові розділи вищої математики-1. Аналітична геометрія	Д.З.	60	1
Історія України	зал.	60	1
Комп'ютерна логіка (К.Р.)	К.Р.	30	1
Комп'ютерна логіка-1. Комп'ютерна логіка.	Екз.	90	1
Програмування	Екз.	240	1
Фізичне виховання (1)	зал.	90	1
Вища математика-2. Диференціальне та інтегральне числення	Екз.	195	2
Дискретна математика	Д.З.	120	2
Іноземна мова- 1. Вступ до загальнотехнічної іноземної мови.	зал.	45	2
Комп'ютерна логіка-2. Комп'ютерна арифметика	Д.З.	90	2
Об'єктно-орієнтоване програмування	Екз.	120	2
Українська мова (за професійним спрямуванням).	зал.	60	2
Фізика	Екз.	270	2
Фізичне виховання (2)	зал.	90	2
Вступ до філософії	зал.	60	3
Додаткові розділи вищої математики-2. Ряди. ТФЗК. Операції	Екз.	60	3
Інженерія програмного забезпечення-1	Екз.	180	3
Комп'ютерна графіка	зал.	90	3
Організація баз даних	Д.З.	120	3
Основи конструювання комп'ютерів-1. Інженерна графіка	зал.	45	3
Психологія	зал.	60	3
Теорія електричних та магнітних кіл	Д.З.	120	3
Теорія ймовірності та математична статистика	Екз.	120	3

Рис. 2.4. Зразок даних предметів

Аналітика «Студент», була штучно розширена додатковою інформацією для варіативності подальшого аналізу. Були додані атрибути, які характеризують студента за статтю, типом навчання (бюджет або контракт), місцем проживання та датою народження. Оскільки таких аналітик не було в початковому файлі, то ці дані (крім статі) було згенеровано випадковим чином, та насичено ними модель. Проте слід зазначити, що така інформація, або будь-яка інша, релевантна до аналізу, може бути додатково завантажена в систему для побудови нових діаграм та збору потрібної статистики. Приклади аналітики «Студент» наведено на рисунку 2.5. В рамках цього проекту згенеровані дані вносяться в систему вручну, а завантажуються лише оцінки студентів по предметам.

Студент	Стать	Тип навчання	Місце проживання	Група	Дата народження
Аксьоненко Ілля Олегович	ч	Бюджет	Не гуртожиток	ІО-62	25.09.1998
Алергі Владислав Віталійович	ч	Бюджет	Гуртожиток	ІО-63	25.02.1998
Андрійчук Дмитро Анатолійович	ч	Бюджет	Не гуртожиток	ІО-64	02.09.1999
Бабко Дмитро Сергійович	ч	Контракт	Гуртожиток	ІО-63	28.09.1999
Балащенко Гліб Ігорович	ч	Контракт	Гуртожиток	ІО-61	16.10.1998
Бандурін Владислав Юрійович	ч	Бюджет	Не гуртожиток	ІО-64	19.10.1998
Бединський Антон Вікторович	ч	Бюджет	Не гуртожиток	ІО-62	17.05.1999
Божок Роман Юрійович	ч	Бюджет	Гуртожиток	ІО-63	27.12.1998
Братун Андрій Юрійович	ч	Бюджет	Гуртожиток	ІО-63	25.04.1998
Бровченко Анастасія Вікторівна	ж	Бюджет	Гуртожиток	ІО-64	24.04.1998
Булах Антон Олегович	ч	Бюджет	Гуртожиток	ІО-63	13.07.1999
Бурбіль Максим Андрійович	ч	Бюджет	Гуртожиток	ІО-62	05.10.1999
Буровецька Ксенія Олександрівна	ж	Бюджет	Гуртожиток	ІО-64	19.01.1999
Валько Антон Вікторович	ч	Бюджет	Не гуртожиток	ІО-61	20.12.1998
Ванярха Сергій Едуардович	ч	Бюджет	Гуртожиток	ІО-63	04.08.1999
Василенко Олександр Вікторович	ч	Контракт	Гуртожиток	ІО-64	20.02.1999
Волошин Микола Віталійович	ч	Контракт	Не гуртожиток	ІО-62	29.06.1998
Воробйов Анатолій Олександрович	ч	Бюджет	Не гуртожиток	ІО-61	21.12.1998
Гаврилюк Олександр Володимирович	ч	Бюджет	Гуртожиток	ІО-64	29.05.1999
Гайдай Анатолій Русланович	ч	Бюджет	Гуртожиток	ІО-63	13.05.1998
Ганжа Денис Віталійович	ч	Бюджет	Не гуртожиток	ІО-64	04.02.1998
Голота Михайло Михайлович	ч	Бюджет	Не гуртожиток	ІО-63	01.03.1999
Гомонець Іван Іванович	ч	Бюджет	Гуртожиток	ІО-62	17.09.1998
Горбов Олексій Юрійович	ч	Бюджет	Гуртожиток	ІО-62	24.03.1999
Гудзенко Олексій Юрійович	ч	Бюджет	Не гуртожиток	ІО-62	26.03.1998
Дарагань Денис Андрійович	ч	Бюджет	Не гуртожиток	ІО-62	15.03.1999
Дем'яненко Максим Олександрович	ч	Бюджет	Гуртожиток	ІО-61	07.08.1999

Рис. 2.5. Зразок даних студентів

При потребі та наявності відповідних даних у вільному доступі модель може бути розширена новими аналітиками: номер залікової книжки, дата вступу, наявність стипендії та будь-якими іншими.

Висновки до розділу 2

В даному розділі було проаналізовано причини вибору інструменту для реалізації системи аналізу даних студентів. Було детально описано хмарну платформу для бізнес-аналізу, яку було вибрано для розробки системи аналізу даних студентів. Розглянуто особливості інструментів, вбудованих в платформу.

Також були зібрані, проаналізовані та описані вхідні дані, необхідні для побудови архітектури системи та подальшого аналізу.

					ІАЛЦ.467800.003 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ПРОЕКТУВАННЯ АНАЛІТИЧНОЇ СИСТЕМИ

3.1. Створення ознак та величин

SAP Analytics Cloud – це хмарне рішення, а це значить, що розробка системи повністю реалізована в веб-інтерфейсі. Доступ до аналітичної платформи здійснюється за веб-лінком, де користувач вводить свій логін та пароль.

Першим кроком в розробці інформаційно-аналітичної системи для аналізу даних студентів є створення бази даних (таблиці), де буде зберігатись уся інформація, яку завантажили або ввели користувачі. В термінології платформи SAP Analytics Cloud така база даних (таблиця) має назву «Модель». Цей об'єкт створюється на основі так званих полів даних, які мають назву ознаки та величини. Вони описують та характеризують будь-яку модель.

При архітектурі моделі даних було вирішено використати 4 ознаки та 1 величину. Саме така кількість об'єктів є достатньою для повноцінного опису системи. Для створення нових об'єктів в системі скористаємось функцією «Створити» - «Величини» (рис. 3.1).

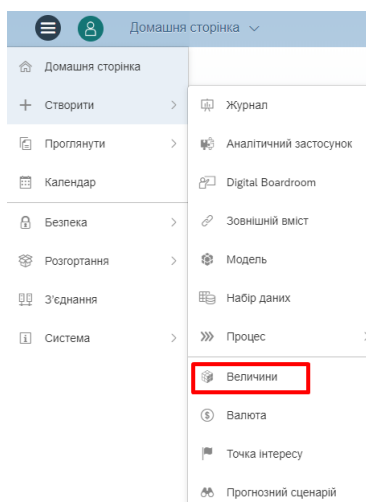


Рис. 3.1. Створення величини

При створенні нової характеристики необхідно явно вказати її тип. В нашому випадку для чотирьох ознак релевантним є універсальний тип. Також необхідно вказати унікальну технічну назву для об'єкту та присвоїти опис (рис. 3.2). Створена ознака, яка буде описувати аналітику «Студент». Також в рамках проекту створено 3 інші ознаки: факультет, спеціальність та предмет.

Створити величину

Тип
Універсальна

Ім'я
Student

Опис
Студент

☐ Увімкнути контроль доступу до даних

☐ Приховати вузли ієрархії для неавторизованих користувачів

☐ Увімкнути відповідальність

☐ Активувати валюту

Створити Скасувати

Рис. 3.2. Створення універсальної величини

В системі також необхідно створити поле для зберігання оцінок та будь-яких показників, які будуть створені в рамках цієї системи. Процес створення схожий зі створенням ознаки, проте в типі величини необхідно вказати тип «Рахунок» (рис. 3.3).

Створити величину

Тип
Рахунок

Ім'я
S_Account

Опис
Показники

☐ Увімкнути контроль доступу до даних

☐ Приховати вузли ієрархії для неавторизованих користувачів

Створити Скасувати

Рис. 3.3. Створення величини з типом «Рахунок»

Після завершення створення необхідних показників та ознак, приступимо до їх насичення даними. Для цього, наприклад, відкриємо створену ознаку «Студент». Як вже було сказано в розділі про роботу з вхідними даними, дана аналітика має декілька властивостей: стать, тип навчання, місце проживання, група та дата народження. Це означає, що кожному студенту можна буде присвоїти дані атрибути. Оскільки всі необхідні дані є в файлі формату «Excel», скопіюємо їх в створений довідник «Студент». Готовий варіант та формат даних для ознаки студент зображено на рисунку 3.4.

ID	Опис	Стать	Тип навчання	Місце проживання	Група	Дата народження
1	Не присвоєно					
2	Аксоненко Ілля Олего...	ч	Бюджет	Не гуртожиток	ІО-62	25.09.1998
3	Алергі Владислав Віта...	ч	Бюджет	Гуртожиток	ІО-63	25.02.1998
4	Андрийчук Дмитро Анат...	ч	Бюджет	Не гуртожиток	ІО-64	02.09.1999
5	Бабо Дмитро Сергію...	ч	Контракт	Гуртожиток	ІО-63	28.09.1999
6	Балащенко Гліб Ігорович	ч	Контракт	Гуртожиток	ІО-61	16.10.1998
7	Бандурін Владислав Ю...	ч	Бюджет	Не гуртожиток	ІО-64	19.10.1998
8	Бедінський Антон Вікт...	ч	Бюджет	Не гуртожиток	ІО-62	17.05.1999
9	Боюк Роман Юрійович	ч	Бюджет	Гуртожиток	ІО-63	27.12.1998
10	Братун Андрій Юрійович	ч	Бюджет	Гуртожиток	ІО-63	25.04.1998
11	Бровченко Анастасія В...	ж	Бюджет	Гуртожиток	ІО-64	24.04.1998
12	Булах Антон Олегович	ч	Бюджет	Гуртожиток	ІО-63	13.07.1999
13	Бурбиль Максим Андрій...	ч	Бюджет	Гуртожиток	ІО-62	05.10.1999
14	Буровецька Ксенія Оле...	ж	Бюджет	Гуртожиток	ІО-64	19.01.1999
15	Валько Антон Вікторович	ч	Бюджет	Не гуртожиток	ІО-61	20.12.1998
16	Ваняра Сергій Едуард...	ч	Бюджет	Гуртожиток	ІО-63	04.08.1999
17	Василенко Олександр ...	ч	Контракт	Гуртожиток	ІО-64	20.02.1999
18	Волошин Микола Вітал...	ч	Контракт	Не гуртожиток	ІО-62	29.06.1998
19	Воробйов Анатолій Оп...	ч	Бюджет	Не гуртожиток	ІО-61	21.12.1998
20	Гаврилюк Олександр В...	ч	Бюджет	Гуртожиток	ІО-64	29.05.1999
21	Гайдай Анатолій Русл...	ч	Бюджет	Гуртожиток	ІО-63	13.05.1998

Рис. 3.4. Вигляд ознаки «Студент»

Для ознаки «Предмет» відповідним чином створено 3 атрибути: форма контролю, години та семестр. Для ознаки «Факультет» та «Спеціальність» були додані релевантні елементи.

В єдиному показнику було створено наступні елементи: «Оцінка», «К-сть предметів», «К-сть спеціальностей», «К-сть факультетів» та «К-сть студентів». В елемент «Оцінка» буде завантажено відповідні дані в підрозділі «Завантаження даних студентів». Для нього було вказано тип виняткової агрегації «AVG» за

величиною виняткової агрегації «Student» та «Subject». Дане рішення дасть змогу продивлятися коректні значення оцінок, а саме середні значення балів по студентам та предметам. Інші створені елементи будуть розраховуватись «на льоту». Тобто користувачам системою непотрібно буде завантажувати або вводити окремі дані для цих елементів. Система власноруч порахує необхідні значення. Для даних елементів було вказано тип виняткової агрегації «COUNT» та відповідні величини виняткових агрегацій. Повну картину налаштувань приведено на рисунку 3.5.

Account						
Account						
ИД элемента	Формула	Тип виняткової агрега...	Величина виняткової ...	Десяткові розряди	Порогове значення	
1 К-сть предметів	1	COUNT	Subject			
2 К-сть спеціальностей	1	COUNT	Specialty			
3 К-сть студентів	1	COUNT	Student			
4 К-сть факультетів	1	COUNT	Faculty			
5 Оцінка		AVG	Student;Subject	2		
6						

Рис. 3.5. Налаштування показників

3.2. Створення моделі даних

Після створення та налаштування всіх необхідних ознак та показників необхідно створити модель даних – об’єкт, який об’єднає в собі всі ознаки та показники. Для цього в інструментальній панелі зліва вибирається функція «Створити» - «Модель» (рис. 3.6). Користувачу пропонується на вибір декілька варіантів створення моделі. В даному випадку потрібно вибрати опцію «Почати з порожньої моделі» (рис. 3.7). Це дозволить нам власноруч створити та налаштувати модель під конкретні потреби та цілі системи.

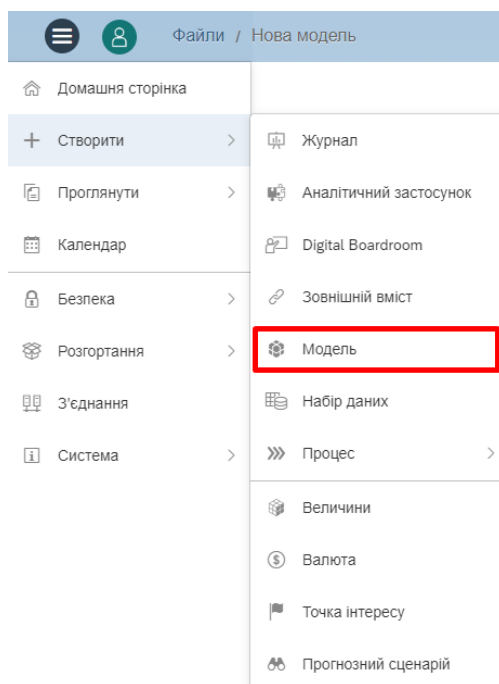


Рис. 3.6. Створення моделі (крок 1)

Опція «Отримати дані з джерела» при створенні моделі дозволяє підключитись до необхідної бази даних, серверу або іншої системи для отримання даних із них. Інша опція «Імпортувати файл з комп'ютеру» дозволяє автоматично створити модель та відразу приступити до її наповнення даними. Однак в поточному випадку модель спочатку налаштовується, а потім в неї завантажуються та вводяться дані.

Виберіть, яким чином почати модель

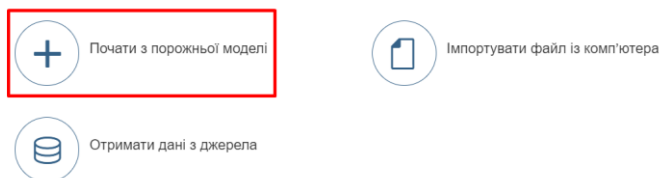


Рис. 3.7. Створення моделі (крок 2)

Наступним кроком в створені та налаштуванні моделі є задання її параметрів. Спочатку модель завжди створюється з типом моделі «Планування». Оскільки в рамках реалізації даної системи функціонал планування не передбачений, цю опцію можна відключити та вказати тип моделі «Аналітика». В даному вікні також задається назва та опис моделі (рис. 3.8).

Параметри моделі

Загальні настройки
Система аналізу дани...

Мова
Українська

Доступ і конфіденці...
Захист даних

Фінансовий час
Вимк.

Планування та діап...
Вимк.

Валюта
USD

Дані та продуктивні...
Продуктивність даних

Інформація про модель

Тип моделі:
Аналітика

Макет моделі:
Рахунок

*Ім'я:
Система аналізу даних студентів

Опис:

Пошук аналітичних даних

Індексувати модель:

OK Скасувати

Рис. 3.8. Параметри моделі

Модель створена та збережена. Залишилось наповнити її ознаками та показниками, які вже були створені на попередньому кроці. Для цього в панелі інструментів зверху в розділі «Редагувати» додаємо в модель усі ознаки та величини, які були створені. Після всіх налаштувань модель буде мати вигляд відповідний рисунку 3.9.

Таким чином було створено та налаштовано модель даних в SAP Analytics Cloud. Іншими словами, було зроблено «розмітку» сховища даних, куди на наступному кроці будуть завантажені оцінки студентів.

Величини				
Ім'я	Опис	Кількість елементів	Кількість ієрархій	Тип
Рахунок				
Account	Account	5	1	Рахунок
Універсальна				
Faculty	Факультет	2	0	Універсальна
Specialty	Спеціальність	2	0	Універсальна
Student	Студент	119	0	Універсальна
Subject	Предмет	63	0	Універсальна

Рис. 3.9. Модель даних

3.3. Завантаження даних студентів в систему

Наступним кроком в створенні сховища даних є його наповнення відповідними даними. В створеній моделі в розділі «Керування даними» необхідно імпортувати в модель файл, який містить дані по оцінкам студентів (рис. 3.10).

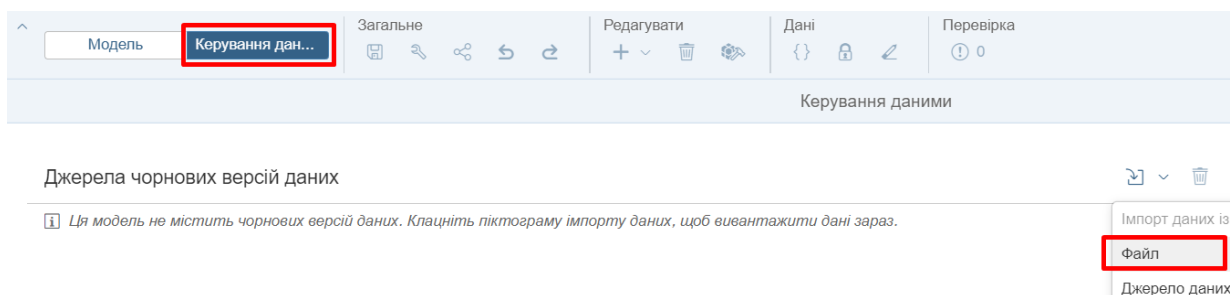


Рис. 3.10. Імпорт даних в модель

Іншим варіантом є імпорт даних із джерела даних: сервера, зовнішньої бази даних чи іншої вихідної системи. Слід зазначити, що при такому підході з'являється можливість з'єднання з даними в режимі реального часу. Така функція дозволяє не зберігати дані в моделі SAP Analytics Cloud, а підключатись моделлю до даних вихідної системи та миттєво відображати їх на дашбордах.

При успішному завантаженні файлу в модель користувачу пропонується продивитись його вміст та зробити необхідні перетворення. Також вказуються опції імпорту даних в модель та налаштовується меппінг полів файлу та ознак і показників моделі. В даному випадку додатково необхідно здійснити транспонування завантажених даних для відповідності з форматом зберігання даних в моделі (рис. 3.11).

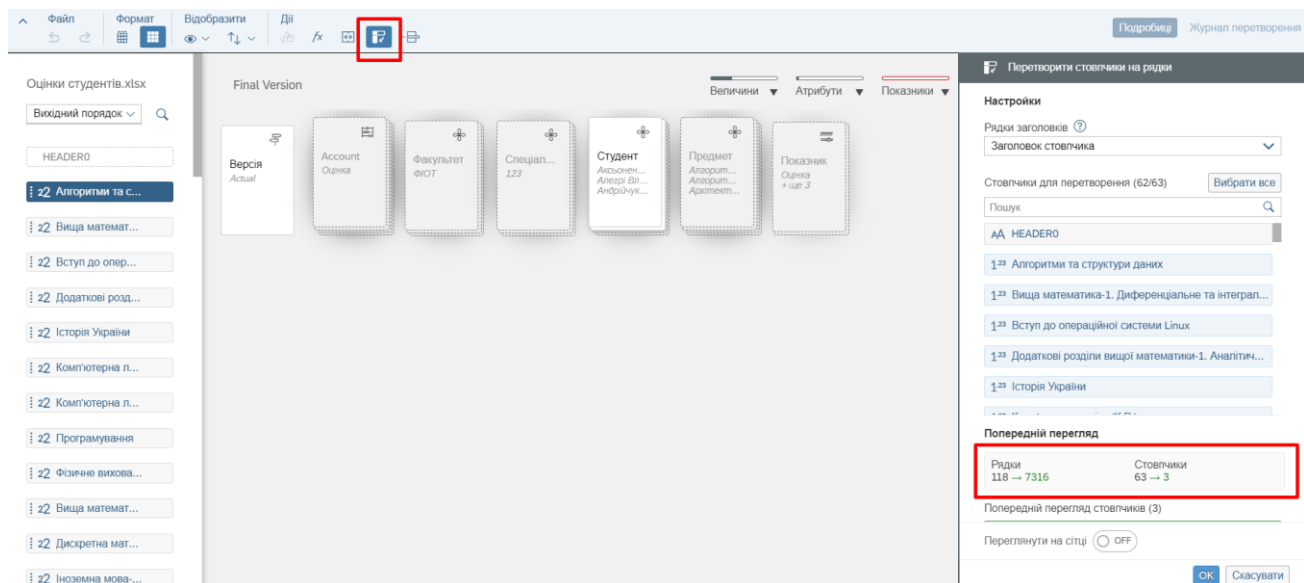


Рис. 3.11. Транспонування даних при імпорті даних в модель

Після транспонування стає можливим здійснити зіставлення (меппінг) даних файлу з полями моделі. Для деяких ознак (факультет та спеціальність) бракує початкових даних, тобто їх неможливо зіставити. Для цього було викорис-

тано функцію «Змінити усталене значення», яка дозволяє напряму вказати на який елемент ознаки буде завантажено усі дані (рис. 3.12). Для ознаки «Факультет» вказано усталене значення «ФІОТ», а для ознаки «Спеціальність» – значення «123» («123» - ключ спеціальності «Комп’ютерна інженерія»).

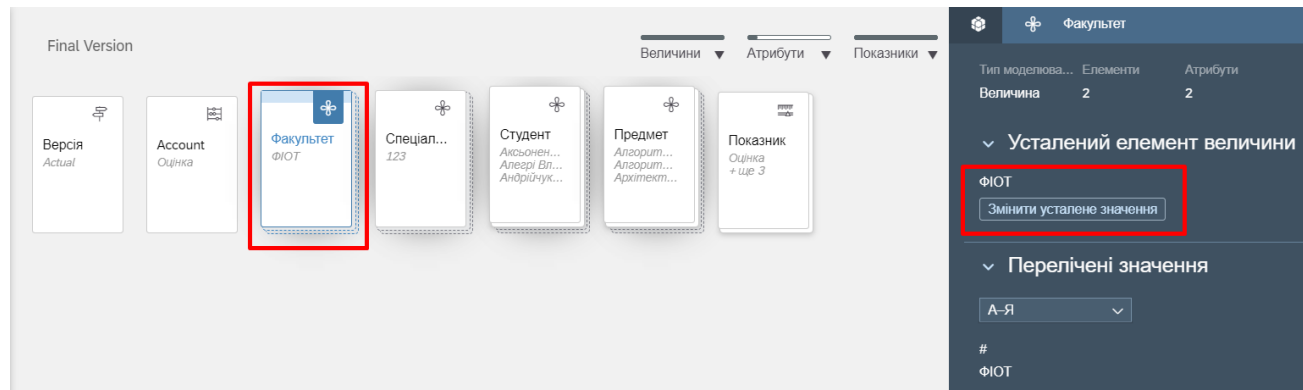


Рис. 3.12. Налаштування імпорту даних в модель

Такий підхід в завантаженні даних дозволяє вести дані в «Excel» файлах або будь-яких інших вихідних системах з меншою деталізацією. При завантаженні цих даних в систему SAP Analytics Cloud користувачам необхідно лише вказати потрібний зріз, куди ці данні будуть записані.

Отже було створено модель із ознаками та показниками. В неї було завантажено дані, а саме оцінки студентів. Таким чином уся необхідна інформація була збережена в хмарі.

3.4. Побудова аналітичного додатку

Наступним кроком проектування аналітичної системи є створення аналітичного додатку, налаштування необхідних розрахунків та побудова візуалізацій.

Аналітичний додаток є розширеним функціоналом продукту SAP Analytics Cloud. За допомогою дизайнера аналітики можна створювати додатки для аналізу і планування даних. На відмінну від звичайних дашбордів, аналітичний додаток дозволяє створювати та налаштовувати гнучку поведінку, фільтрацію та взаємозв'язок візуальних елементів.

Аналітичний додаток створюється в спеціальному середовищі розробки SAP Analytics Cloud, яке називається дизайнером аналітики. Дане середовище об'єднує в собі звичайні можливості побудови дашборду та спеціалізовані можливості по написанню скриптів для кожного об'єкту. Так, наприклад, для стовпчастої діаграми можна написати подію, яка буде викликатись при натисканні на будь-який елемент цієї діаграми. Це може бути фільтрація інших об'єктів по вибраному зрізу даних (зв'язаний аналіз), перехід на іншу сторінку, відкриття спливаючого вікна та інше.

Для початку роботи в середовищі дизайнера аналітики необхідно створити новий аналітичний додаток. Йому було присвоєно технічне ім'я та опис, щоб полегшити його пошук по всій системі.

Першим кроком в налаштуванні додатку є підключення необхідних даних для візуалізації. В нашому випадку в додаток додається модель, створена на попередніх етапах розробки. Далі користувачу пропонується на вибір безліч можливостей для дизайну свого додатку та подальших налаштувань.

Перед створенням необхідних діаграм, таблиць та об'єктів було визначено та реалізовано додаткові розрахунки, які будуть задіяні для коректних відображень результатів. Розрахунки на рівні аналітичного додатку умовно можна поділити на два типи – це розрахункові ознаки та розрахункові показники. Перший тип необхідний для створення нових зрізів даних, а другий – для створення та розрахунків безпосередньо даних. Завдяки розрахунковим показникам можна

налаштовувати будь-які математичні операції, використовувати умови або просто обмежувати існуючий показник по необхідному зрізу даних.

В рамках цього проекту було створено наступні розрахункові показники:

- «Кількість груп»;
- «Кількість оцінок»;
- «Кількість годин предмету»;
- «Кількість оцінок від 85 до 100»;
- «Кількість оцінок менше 75»;
- «Кількість оцінок менше 75 (менше 3)»;
- «Кількість предметів у семестрі (середня)»;
- «Кількість студентів у групі (середня)»;
- «Середня оцінка потоку»;
- «% Оцінок від 85 до 100»;
- «R_Marks».

Створені показники дають змогу аналізувати додаткові дані та збирати необхідну в рамках проекту статистику. Приклад обчислення розрахункового показника зображено на рисунку 3.13.

Редактор обчислень

Тип

Обчислюваний показник

Ім'я

К-сть оцінок менше 75 (менше 3)

Редагувати формулу

1 IF ([К-сть оцінок менше 75] <= 3 ; [К-сть оцінок менше 75] ; NULL)

Формат

Функції формули

Функції

IF()

ABS()

LOG()

LOG10()

INT()

FLOAT()

DOUBLE()

POWER()

GrandTotal()

%GrandTotal()

OK Скасувати

Рис. 3.13. Налаштування показника «Кількість оцінок менше 75 (менше 3)»

Також було створено розрахункові ознаки:

- «Розбаловка»;
- «Вимір оцінки»;
- «Вимір оцінки (по семестру)»;
- «R_Student»;
- «R_Group».

Створені розрахункові ознаки вводять додаткові можливості для аналізу. Так, наприклад, ознака «Розбаловка» дасть можливість проаналізувати розподіл оцінок по предметам або студентам. Налаштування розрахункової ознаки «Розбаловка» наведено на рисунку 3.14.

Редактор обчислень

Тип
Величина на основі показника

Властивості

Показник
Оцінка

☐ Використовувати значення показників як елементи величин

Ім'я елемента A

B

C

D

E

F

+ Додати елемент

Контекст величини
Версія, Предмет, Студент

Контекст фільтра

Величини
Клацніть, щоб вибрати величину...

+ Додати величину

Ім'я
Розбаловка

Значення величин

≥ 95

≥ 85

≥ 75

≥ 65

≥ 60

≥ Нижня межа

≤ 100

< 95

< 85

< 75

< 65

< 60

Значення або елементи керування введенням
Клацніть, щоб вибрати значення...

OK
Скасувати

Рис. 3.14. Налаштування ознаки «Розбаловка»

Таким чином, було створено та налаштовано розрахункові показники та ознаки. Перевагою такого підходу є те, що ці характеристики та нові дані не потрібно завантажувати в систему, щоб їх продивитись та проаналізувати. Більше того, дані по новоствореним розрахункам фізично не зберігаються в моделі. Усі розрахунки даних відбуваються автоматично на льоту в режимі онлайн при відкритті користувачем аналітичного додатку.

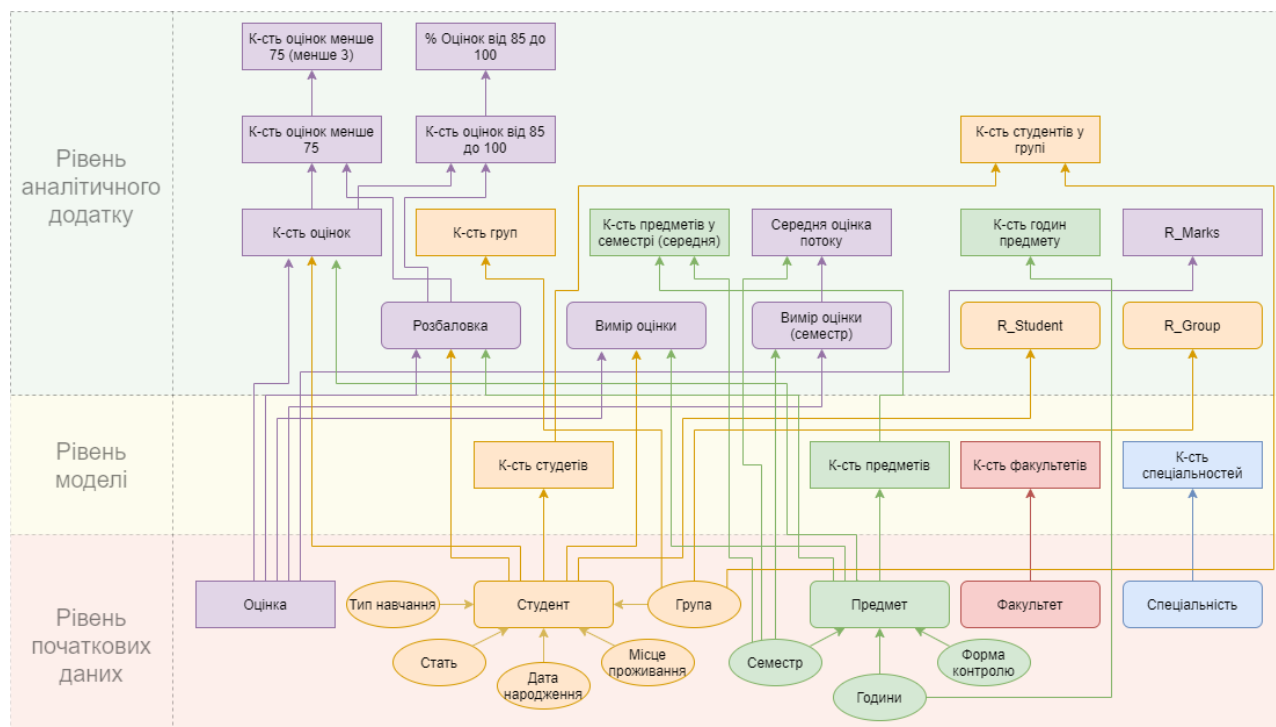


Рис. 3.15. Взаємозв'язок ознак та показників аналітичної системи

На рисунку 3.15 зображено повна архітектура бази даних та взаємозв'язки між створеними ознаками та показниками. На рівні початкових даних ознаки та показники фізично зберігають дані (введені або завантажені). На рівні моделі відбуваються розрахунки деяких показників, проте вони вже фізично нічого не зберігають, а задаються формулами та умовами. Рівень аналітичного додатку відповідає за розрахунки в відповідному середовищі, проводячи їх «на льоту» без фізичного зберігання даних.

Після створення необхідних розрахункових показників та ознак на рівні аналітичного додатку було створено панелі та спливаючі вікна для логічного групування елементів візуалізації. Побудовано 3 виринаючих вікна:

- «rporDetails» – вікно, яке призначене для детального аналізу розподілу оцінок;

- «popupStudent» – вікно, яке призначене для детального аналізу даних по вибраному студенту;
- «popupSubject» – вікно, яке призначене для детального аналізу даних по вибраному предмету.

Виринаючі вікна відкриваються по кліку на відповідний елемент діаграми. Наприклад, для відкриття вікна з детальним аналізом студента, користувачеві потрібно вибрати студента на будь-якій діаграмі чи таблиці, де цей студент присутній. Скрипт, який дозволить виконувати дану операцію, розглянутий в деталях далі.

Також створено 2 панелі для групування об'єктів на головному екрані аналітичного додатку:

- panelMain – логічне групування елементів першої сторінки головного екрану додатку (сторінка огляду статистики);
- panelSecond – логічне групування елементів другої сторінки головного екрану додатку (сторінка аналізу оцінювання).

Всі інші об'єкти будуть розташовані на стандартній панелі, яка називається «Полотно». Ці елементи візуалізації будуть створюють умовний бекграунд для всього додатку. Наприклад, на полотні розташовано заголовок діаграми, плашки для фільтрації та інші інформаційні елементи, які не потребують того, щоб їх відображення змінювалось в ході роботи аналітичного застосунку.

Наступним кроком в розробці аналітичного додатку є створення та наповнення його контентом, тобто побудова та налаштування віджетів, діаграм та таблиць. Для головної сторінки «Загальний огляд» було створено та налаштовано наступні діаграми:

- «К-сть студентів по групам»;
- «К-сть студентів по статі»;

- «К-сть студент по типу навчання та місцю проживання»;
- «Динаміка к-сті предметів по семестрам»;
- «К-сть предметів по формам контролю».

Налаштування та структура діаграм приведено на рисунку 3.16 на прикладі діаграми «Динаміка к-сті предметів по семестрам». В показники діаграми додано елемент «К-сть предметів», в величини – ознаку «Семестр», а в розділ кольору – атрибут «Форма контролю». Був вибраний складений та стовпчастий тип діаграми, що дозволило відобразити дані з необхідною деталізацією. Таким чином користувачі матимуть змогу переглянути динаміку кількості предметів по семестрам та відслідковувати їх по формам контролю. Наприклад, з легкістю можна зрозуміти в яких семестрах студентам було запропоновано виконання курсових робіт. На дану діаграму було також додано посиальну лінію, яка відображає середнє значення кількості предметів по семестрам.

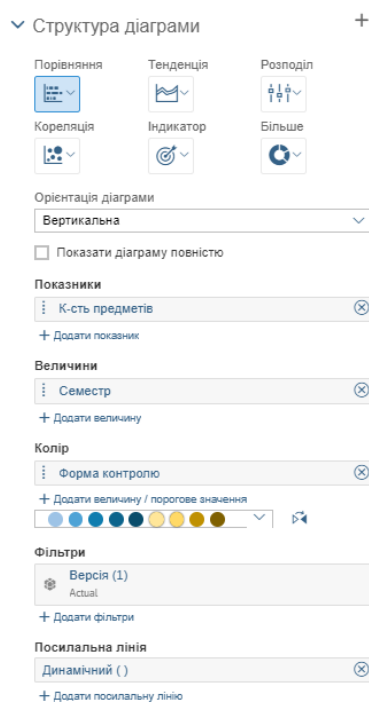


Рис. 3.16. Структура діаграми «Динаміка к-сті предметів по семестрам»

На головному екрані також було створено шість плашок для аналізу ключових показників ефективності навчання:

- «К-сть спеціальностей»;
- «К-сть груп»;
- «К-сть студентів»;
- «К-сть предметів»;
- «К-сть студентів у групі (середня)»;
- «К-сть предметів у семестрі (середня)»;

В діаграми було додано відповідні показники, які було створено на минулому кроці. Також налаштоване форматування для приємного вигляду всіх діаграм.

Для другої сторінки аналітичного застосунку «Аналіз оцінок» було створено ряд наступних діаграм:

- «Середній бал по групам»;
- «Середній бал по студентам»;
- «Претенденти на диплом з відзнакою»;
- «Розподіл оцінок»;
- «Залежність оцінок від тривалості предмету»;
- «Динаміка к-сть предметів та оцінок по семестрам»;
- «Середня оцінка по типу навчання та місцю проживання».

На рисунку 3.17 приведено приклад структури та налаштувань однієї з діаграм даної сторінки. Для діаграми вибрана категорія «Кореляція» та тип «Діаграма розсіювання», що дає змогу відстежити залежність показника, який додано в вісь X, від показника, який додано в вісь Y. Також додано та налаштовано дві посилальні лінії, які відображають середнє значить кожного із показників.

Включена функція «Інтелектуального групування», що автоматично аналізує точки даних на діаграмі та групує їх на основі схожості властивостей.

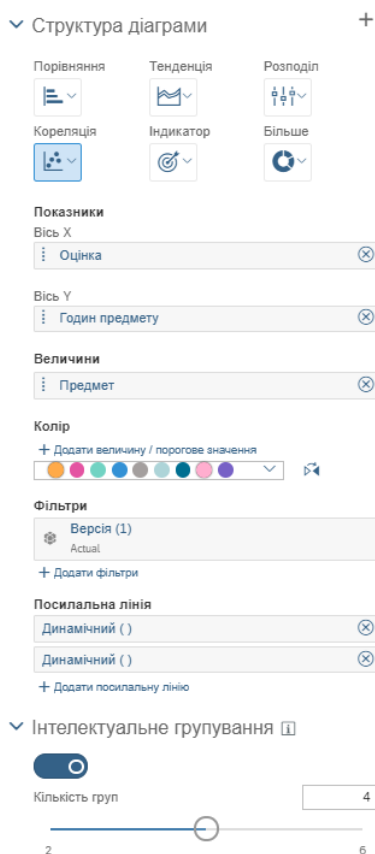


Рис. 3.17. Структура діаграми «Залежність оцінки від тривалості предмету»

На сторінці з детальним аналізом приводиться таблиця зі зрізом усіх студентів та всіх предметів. Налаштування таблиці аналогічно налаштуванню діаграми. Також на сторінці присутня діаграма «Середній бал по семестрам» для можливості фільтрації таблиці по кліку на дану діаграму (зв'язаний аналіз), яка дописано скриптом.

Усі елементи візуалізації унікальне технічне ім'я для того, щоб до них можна було звертатись при написанні функцій та скриптів. Нижче наведено пе-

релік усіх віджетів аналітичного додатку по сторінкам та виринаючим вікнам (рис. 3.18-3.20).

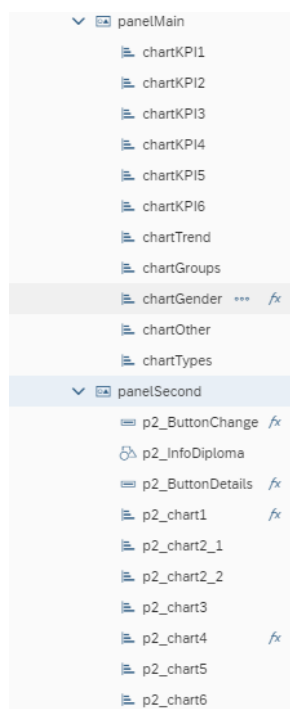


Рис. 3.18. Перелік об'єктів першої та другої сторінки аналітичного додатку

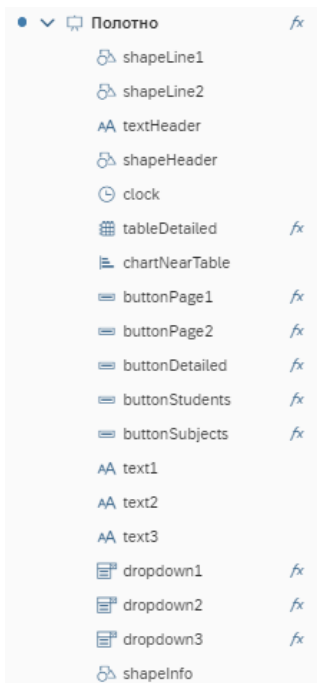


Рис. 3.19. Перелік об'єктів полотна аналітичного додатку

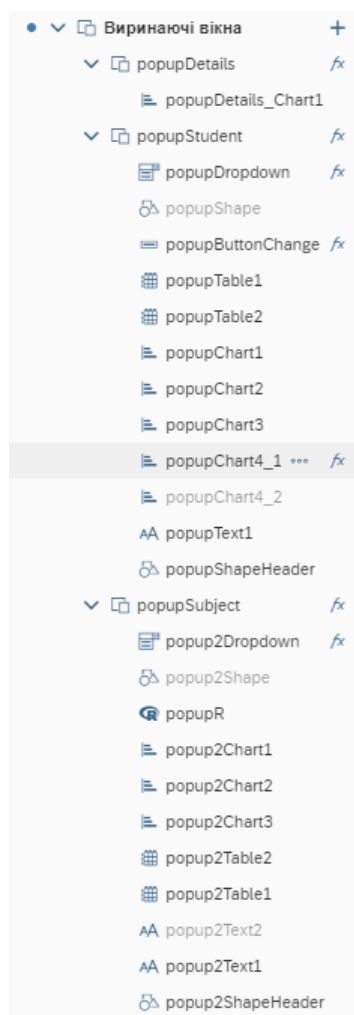


Рис. 3.20. Перелік об'єктів виринаючих вікон

Для виринаючого вікна з детальним аналізом студента було створено та налаштовано наступні діаграми та таблиці:

- «Середній бал студента»;
- «Середній бал потоку»;
- «Бали студента»;
- «Середній бал по семестрам (Відображення №1)»;
- «Середній бал по семестрам (Відображення №2)»;
- «Оцінки по предметам»;

- «Детальна інформація».

Діаграма, яка показує середній бал студента по семестрам має два варіанти відображення. Перший варіант відображає динаміку оцінювання студента, другий варіант – порівнює динаміку оцінювання студента з динамікою оцінювання всього потоку. Налаштоване зручне переключення режиму перегляду за допомогою кнопки та простого скрипту, який був написаний для події натискання на кнопку (рис. 3.21).

```

popupButtonChange – onClick
Викликається, коли користувач натискає кнопку.

function onClick() : void

1
2 if (this.getText() === 'Відображення №2'){
3     this.setText('Відображення №1');
4     popupChart4_2.setVisible(true);
5     popupChart4_1.setVisible(false);
6 }
7 else{
8     this.setText('Відображення №2');
9     popupChart4_2.setVisible(false);
10    popupChart4_1.setVisible(true);
11 }
12

```

Рис. 3.21. Вигляд лістингу скрипта натискання кнопки

Для виринаючого вікна з детальним аналізом предмету було створено та налаштовано наступні діаграми та таблиці:

- «Середній бал по предмету»;
- «Середній бал усіх предметів»;
- «Бали по предмету»;
- «Student marks by group»;
- «Оцінки предмету по студента»;
- «Детальний аналіз предмету».

Діаграма «Student marks by group» - це спеціальний віджет, написаний на мові програмування R. При написанні коду був використаний пакет ggplot2, необхідний для побудови та відображення діаграми вибраного типу (рис. 3.22). Такий підхід дозволив побудувати специфічну діаграму з користувацькими налаштуваннями.

```

Редактор*
1 # Simulates data and visualizes them as a Jitter Plot
2 # load packages
3 library(ggplot2)
4
5 # assign data
6 set.seed(123)
7
8 newData <- Final_Version
9 print(newData)
10
11 # add styling
12 outputGraphics <- ggplot(newData, aes(x = newData$R_Group, y = newData$Marks, colour = newData$R_Group, size=1, alpha=0.5))
13 + geom_jitter() + ggtitle("Student marks by group") + xlab("Group") + ylab("Marks")
14
15 # output
16 outputGraphics + theme_bw() + theme(legend.position = "none", aspect.ratio = 0.5,
17                                     panel.grid.major = element_blank(),
18                                     panel.grid.minor = element_blank(),
19                                     plot.title = element_text(hjust = 0.5),
20                                     text = element_text(size = 15))
21

```

Рис. 3.22. Вигляд лістингу скрипта на мові R

На обидва виринаючих вікна було додано по випадаючому списку. Даний об'єкт налаштовано як елемент для фільтрації, тобто користувачі матимуть змогу вибрати будь-який елемент із списку для подальшої фільтрації поточного виринаючого вікна. Дані випадаючі списки заповнюються автоматично при ініціалізації аналітичної програми (рис. 3.23). Коли користувач вибирає елемент із випадаючого списку викликається подія скрипту «onSelect» для даного об'єкту (рис. 3.24).

```

scripts – feelDropdownSubjects
Заполнить выпадающий список всеми предметами

function feelDropdownSubjects() : void

1 // заполнение выпадающего списка всеми элементами измерения "Предмет"
2
3 var subjects = popup2Chart1.getDataSource().getMembers("Subject");
4
5 //console.log(subjects);
6
7
8 for (var i=0; i<subjects.length; i++){
9     //console.log(subjects[i].id);
10
11     if (subjects[i].id!="#"){
12         popup2Dropdown.addItem(subjects[i].id);
13     }
14 }
15

```

Рис. 3.23. Вигляд функції заповнення випадаючого списку

```

popup2Dropdown – onSelect
Викликається, коли користувач вибирає елемент із розкритого списку.

function onSelect() : void

1 console.log('Popup2 Dropdown - onSelect() function');
2
3 // скривається фігура, которая засеривала всю страницу
4 popup2Shape.setVisible(false);
5
6 // выбранный элемент выпадающего списка записывается в переменную
7 var current_subject = this.getSelectedKey();
8 console.log(current_subject);
9
10 // все объекты на странице (кроме общего ср. балла) фильтруются по элементу, выбранному в выпадающем списке
11 popup2Chart1.getDataSource().setDimensionFilter("Subject", current_subject);
12 popup2Chart3.getDataSource().setDimensionFilter("Subject", current_subject);
13
14 popup2Table1.getDataSource().setDimensionFilter("Subject", current_subject);
15 popup2Table2.getDataSource().setDimensionFilter("Subject", current_subject);
16
17 popupR.getDataFrame("Final_Version").getDataSource().setDimensionFilter("Subject", current_subject);
18
19 // если текст длинный - используется заголовок с меньшим шрифтом (чтобы текст поместился)
20 // иначе - используется обычный большой заголовок
21
22 console.log('Function workSubjectLength');
23
24 if (current_subject.length > 50){
25     popup2Text1.setVisible(false);
26     popup2Text2.setVisible(true);
27     popup2Text2.applyText(current_subject);
28 }
29 else{
30     popup2Text2.setVisible(false);
31     popup2Text1.setVisible(true);
32     popup2Text1.applyText(current_subject);
33 }
34

```

Рис. 3.24. Вигляд функції фільтрації віринаючого вікна

Останнім кроком в побудові аналітичного застосунку стало налаштування дизайну: застосування стилів та форматування, додання динамічного розташування та розмірів об'єктів.

Насамперед, було налаштовано порогове значення для показника «Оцінка» (рис. 3.25). Дана забаганка необхідна для візуально-зрозумілого відображення та класифікації оцінок студентів.

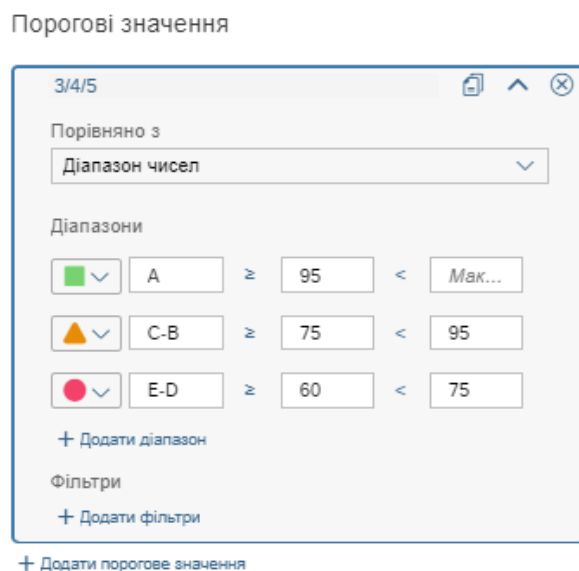


Рис. 3.25. Налаштування порогового значення для показника «Оцінка»

Також для всіх діаграм було налаштовано необхідні стилі відображення. Вказано шрифти, їх розміри та кольори, задані формати чисел та формати самих діаграм. Там, де це необхідно, була сформована і застосована палітра кольорів.

Окремо було написано скрипт, який динамічно розташовує об'єкти по аналітичному застосунку в залежності від розмірів дисплею користувачам (рис. 3.26). Тобто, було налаштовано динамічне масштабування програми в залежності від розширення екрану, на якому цей додаток буде запущений.

```

Application – onInitialization
Called when the analytic application has finished loading.

function onInitialization() : void

7 // узнаем высоту и ширину экрана при первом запуске приложения
8 var height = this.getInnerHeight();
9 var width = this.getInnerWidth();
10 console.log([height, width]);
11
12 var border = 32;
13 var smallBorder = 16;
14
15 // изменяем размеры бекграунда заголовка в зависимости от ширины экрана
16 shapeHeader.getLayout().setWidth(width);
17
18 // узнаем длину текста в заголовке и конвертируем его в integer
19 var textWidthInt = ConvertUtils.stringToInteger(ConvertUtils.numberToString(textHeader.getLayout().getWidth().value));
20 // по формуле находим Left(x), где должен стоять текст и конвертируем его (x) в integer
21 var textLeft = ConvertUtils.stringToInteger(ConvertUtils.numberToString((width.value-textWidthInt)/2));
22 console.log(['Text left:', textLeft]);
23 // ставим текст заголовка по центру экрана
24 textHeader.getLayout().setLeft(textLeft);
25
26 var headerHeight = shapeHeader.getLayout().getHeight().value;
27 console.log(['Header Height: ', headerHeight]);
28
29 // изменяем размер основной панели
30 panelMain.getLayout().setWidth(width);
31 panelMain.getLayout().setHeight(height.value-headerHeight);
32
33 // рассчитываем динамическую ширину объектов
34 var objectWidth = (width.value-border*5)/4;
35 // конвертируем number в integer для корректного применения в функциях
36 var objectWidthInt = ConvertUtils.stringToInteger(ConvertUtils.numberToString(objectWidth));
37 var smallObjectWidth = ConvertUtils.stringToInteger(ConvertUtils.numberToString((objectWidthInt-border/2)/2));
38 var smallObjectHeight = ConvertUtils.stringToInteger(ConvertUtils.numberToString((objectWidthInt-border)/3));
39 console.log(objectWidthInt);
40
41 // настраиваем кнопку
42 buttonGradesPage.getLayout().setLeft(border*3);
43 buttonGradesPage.getLayout().setTop(objectWidthInt+border);
44 //buttonGradesPage.getLayout().setHeight(objectWidthInt);
45 buttonGradesPage.getLayout().setWidth(objectWidthInt-border*4);
46
47 // настраиваем кнопку (детальный анализ)
48 buttonDetailed.getLayout().setLeft(border*3);
49 buttonDetailed.getLayout().setTop(objectWidthInt-border);
50 //buttonGradesPage.getLayout().setHeight(objectWidthInt);
51 buttonDetailed.getLayout().setWidth(objectWidthInt-border*4);

```

Рис. 3.26. Вигляд функції динамічного масштабування аналітичного додатку

Таким чином, було спроектовано та побудовано аналітичний застосунок. Налаштовано необхідні ознаки та показники на рівні додатку. Створено елементи візуалізацій та написано скрипти для їх гнучкої взаємодії.

Висновки до розділу 3

Під час розробки третього розділу було побудовано архітектуру інформаційно-аналітичної системи та завантажено дані для аналізу.

Було створено ознаки та величини, які зберігають в собі інформацію та відіграють роль полів таблиці в базі даних. Побудовано модель даних на основі створених аналітик. Додатково створені нові аналітики, які розраховуються автоматично.

Останнім кроком розробки системи стало створення аналітичного додатку. За допомогою функціоналу дизайнера аналітики було створено додаткові показники, необхідні для аналізу, побудовано діаграми та таблиці, написано скрипти для гнучкої взаємодії елементів візуалізацій.

РОЗДІЛ 4 ВІЗУАЛІЗАЦІЯ ДАНИХ АНАЛІТИЧНОЇ СИСТЕМИ

4.1. Сторінка аналітичного додатку «Загальний огляд»

При відкритті аналітичного додатку «Система аналізу даних студентів» користувачам насамперед пропонується загальний огляд статистичних даних на першій сторінці (рис. 4.1). Приводяться діаграми, які відображають розподіл студентів по групах, статі, типу навчання там місцю проживання. В нижній частині сторінки аналізуються предмети по формам контролю та динаміка їх кількості протягом семестру. Зліва внизу відображаються ключові показники ефективності в вигляді числових діаграм.



Рис. 4.1. Сторінка аналітичного додатку «Загальний огляд»

Інформаційна панель з розкривними списками та кнопками призначена для фільтрації всіх віджетів та зміни відображення відповідно. Таким чином, користувач може вибрати з розкривного списку, наприклад, іншу спеціальність, щоб продивитись відповідні дані на головному екрані аналітичного додатку. За допомогою кнопок можна з легкістю перемикатись між сторінками та відкривати виринаючі вікна.

Одна з властивостей кругових діаграм – це відображення не тільки абсолютних значень, а й відсоткового співвідношення. При тому, по центрі виводиться загальна кількість значень, що робить дану діаграму підходящою для усестороннього розгляду. На прикладі діаграми «К-сть предметів по формах контролю» (рис. 4.2) користувачі можуть зрозуміти сумарну кількість предметів, кількість предметів по формах контролю, а також їх відсоткове співвідношення.

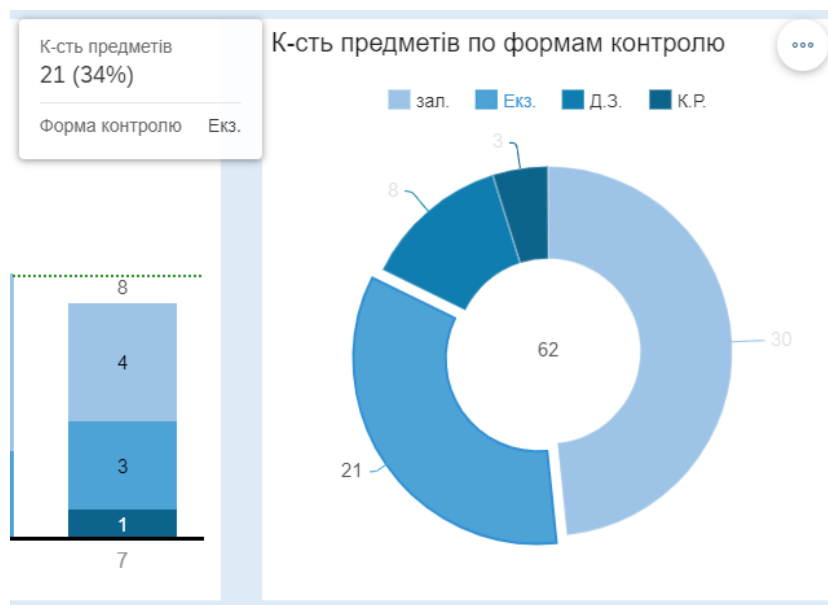


Рис. 4.2. Діаграма «К-сть предметів по формах контролю»

Отже, сторінка аналітичного застосунку «Загальний огляд» дає змогу зібрати та проаналізувати статистичні дані по студентам та предметам. Налашто-

вані фільтри дають змогу вибрати необхідний для аналізу зріз даних. Усі графіки та об'єкти сторінки миттєво підлаштовуються під зміни.

4.2. Сторінка аналітичного додатку «Аналіз оцінок»

При натисканні кнопки «Аналіз оцінок» відображення аналітичного додатку відразу зміниться та перемкнеться на відповідну сторінку (рис. 4.3).

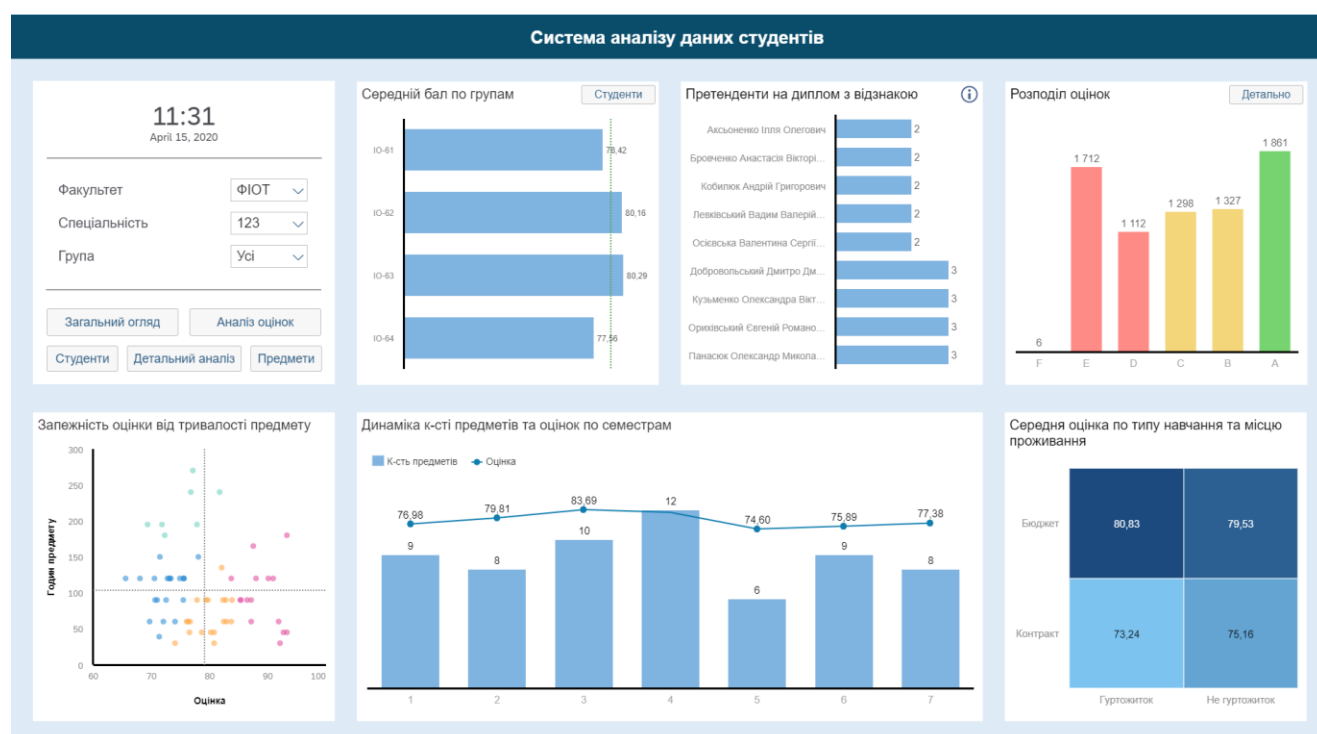


Рис. 4.3. Сторінка аналітичного додатку «Аналіз оцінок»

Дана сторінка спеціалізується на огляді та аналізі оцінок та пошуку будь-яких залежностей цього показника. Інструментальна панель зліва залишається для можливостей фільтрації та подальшого зміни відображення.

До діаграми «Середній бал по групам» додана кнопка «Студенти», при натисканні якої в вісь діаграми замість груп додаються студенти (рис. 4.4). Так як

загальна кількість студентів занадто велика, автоматично застосовується опція фільтрації десяти найкращих студентів за середнім балом. Однак користувач власноруч може змінити цю опцію на будь-яку іншу або навіть відобразити всіх студентів. Також кнопка «Студенти» змінилась кнопкою «Групи» для можливості відновлення стандартного відображення.



Рис. 4.4. Діаграма «Середній бал по студентам»

Діаграма «Претенденти на диплом з відзнакою», можливо, є самою корисною в усьому аналітичному додатку. На ній виводяться тільки ті студенти, які мають менше трьох оцінок нижче 75 балів та відсоток оцінок від 85 до 100 більше 75. Саме це є двома умовами на диплом із відзнакою. Кількість оцінок менше 75 приводиться в вигляді стовпчастої діаграми, а відсоток оцінок від 85 до 100 – в виринаючій підказці, яка з'являється при наведенні миші на будь-якого студента. При кліку на студента відкривається виринаюче вікно, яке дає змогу детально проаналізувати даного студента та зрозуміти які предмети йому потрібно перездати, щоб отримати диплом з відзнакою.

Діаграма «Розподіл оцінок» дозволяє проаналізувати розповсюдження всіх оцінок. Для детального аналізу пропонується натиснути кнопку «Детально». З'являється виринаюче вікно з діаграмою, яка показує кількість оцінок по кожній із оцінок (рис. 4.5). Розподіл дає зрозуміти, що такі оцінки як «60» та «95» значно переважають інші. Цікавим також є те, що кількість округлених оцінок значно більша за будь-які інші бали.



Рис. 4.5. Діаграма «Детальний розподіл оцінок»

Діаграма «Залежність оцінки від тривалості предмету» зручно та наочно демонструє повну картину оцінок по всім предметам в залежності від тривалості предметів. Кожна точка – це предмет, при наведенні на який відображається підказка з даними по предмету (рис. 4.6). Діаграма використовує автоматичне групування елементів за схожістю їх властивостей. Таким чином було створено 4 групи, які класифікують всі предмети за категоріями. Для подальшого детального аналізу предмету можна натиснути на будь-яку точку (предмет) на діаграмі – відкриється виринаюче вікно з детальними інформаційно-аналітичними даними по вибраному предмету.

Також дана точкова діаграма дає зрозуміти, що розподіл оцінок хаотичний, а явної залежності тривалості предмету від оцінки не прослідковується. Однак, наочно видно предмети з занадто низькими та високими показниками оцінювання. Наприклад, предмет «Комп'ютерні системи» з середнім балом 65,61 явно має проблеми з рейтинговою системою оцінювання та потребує подальшого перегляду та аналізу причин виникнення такого зміщення. В свою чергу, посильні пунктирні лінії на діаграмі дають змогу оцінити середнє значення оцінок по всіх предметах та середнє значення тривалості предмету. Таким чином діаграма поділяється на чотири умовних сектори, які також можуть класифікувати предмети в залежності від зони їх розташування.

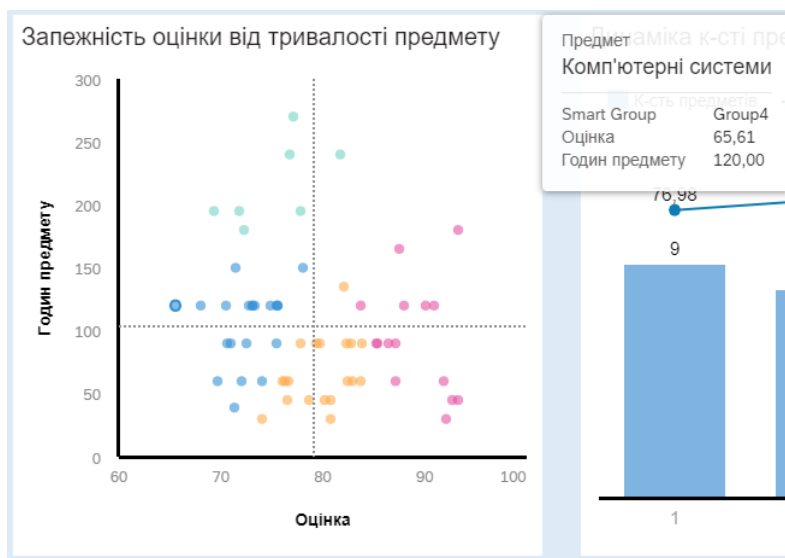


Рис. 4.6. Діаграма «Залежність оцінки від тривалості предмету»

Отже, сторінка аналітичного застосунку «Аналіз оцінок» дає змогу проаналізувати оцінки по різних зрізах даних. Приводяться діаграми, які знаходять різноманітні залежності, підказують інші факти та описують розподіл оцінок. За допомогою скриптів на діаграми налаштовано гнучкий перехід до подальшого детального аналізу в виринаючих вікнах.

4.3. Сторінка аналітичного додатку «Детальний аналіз»

При натисканні кнопки «Детальний аналіз» відображення аналітичного додатку відразу зміниться та перемкнеться на відповідну сторінку (рис. 4.7).

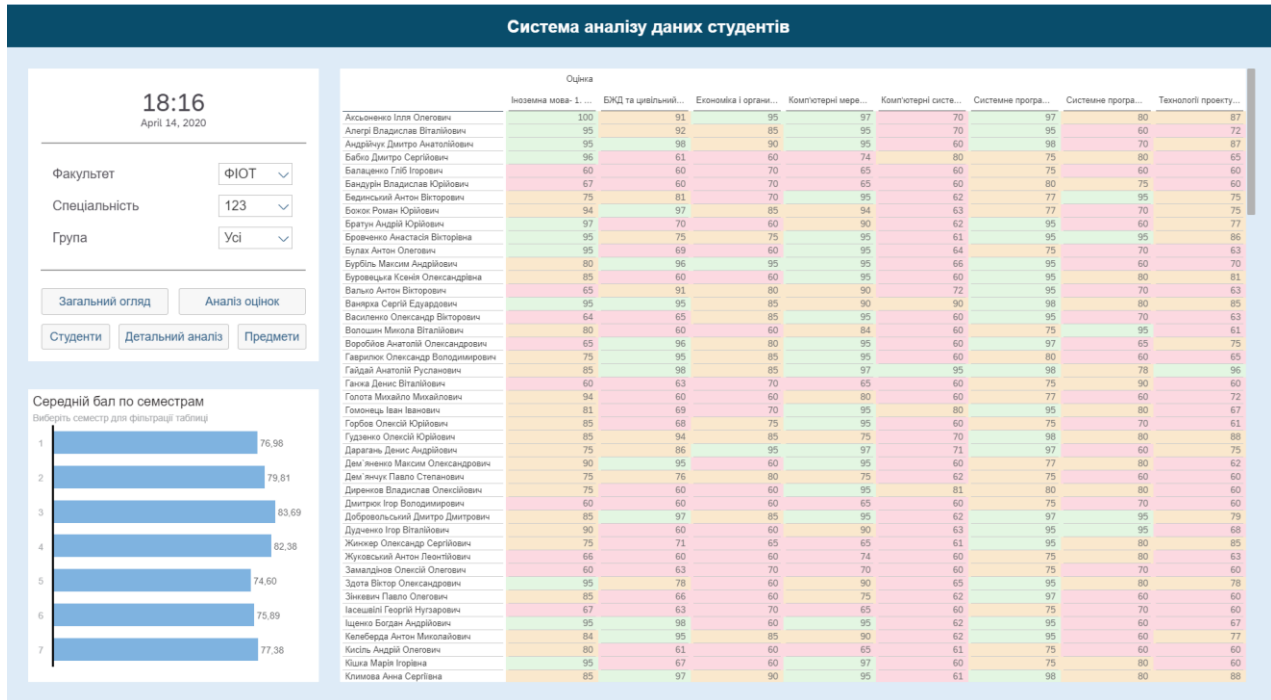


Рис. 4.7. Сторінка аналітичного додатку «Детальний аналіз»

На даній сторінці приводиться таблиця, яка відображає оцінки всіх студентів по всім вибраним предметам. Даний аналіз необхідний для пошуку патернів та відслідковування оцінок по конкретному студенту чи предмету. Для таблиці задіяно порогові значення, що дало змогу автоматично зафарбувати оцінки в відповідний колір і полегшило виявлення патернів. Також користувач може вибрати будь-якого студента в таблиці для відкриття виринаючого вікна з детальним оглядом даних цього студента.

Діаграма «Середній бал по семестрам» відображає відповідні дані та є імпровізованим фільтром для таблиці. Тобто, користувач може вибрати будь-який стовпчик на діаграмі, а дані на таблиці будуть автоматично відфільтровані відповідним чином.

Для повернення на будь-яку із попередніх сторінок аналітичного застосунок потрібно натиснути кнопки, які приведені на інформаційній панелі зліва. Фільтрація таблиці також можлива за допомогою фільтрів на панелі.

4.4. Виринаюче вікно аналітичного додатку «Аналіз студента»

Виринаюче вікно «Аналіз студента» (рис. 4.8) може бути відкрите в декілька способів: натисканням кнопки «Студенти» на інформаційній панелі з будь-якого місця програми або вибором ознаки «Студент» на будь-якій діаграмі чи таблиці.

Аналіз успішності студента

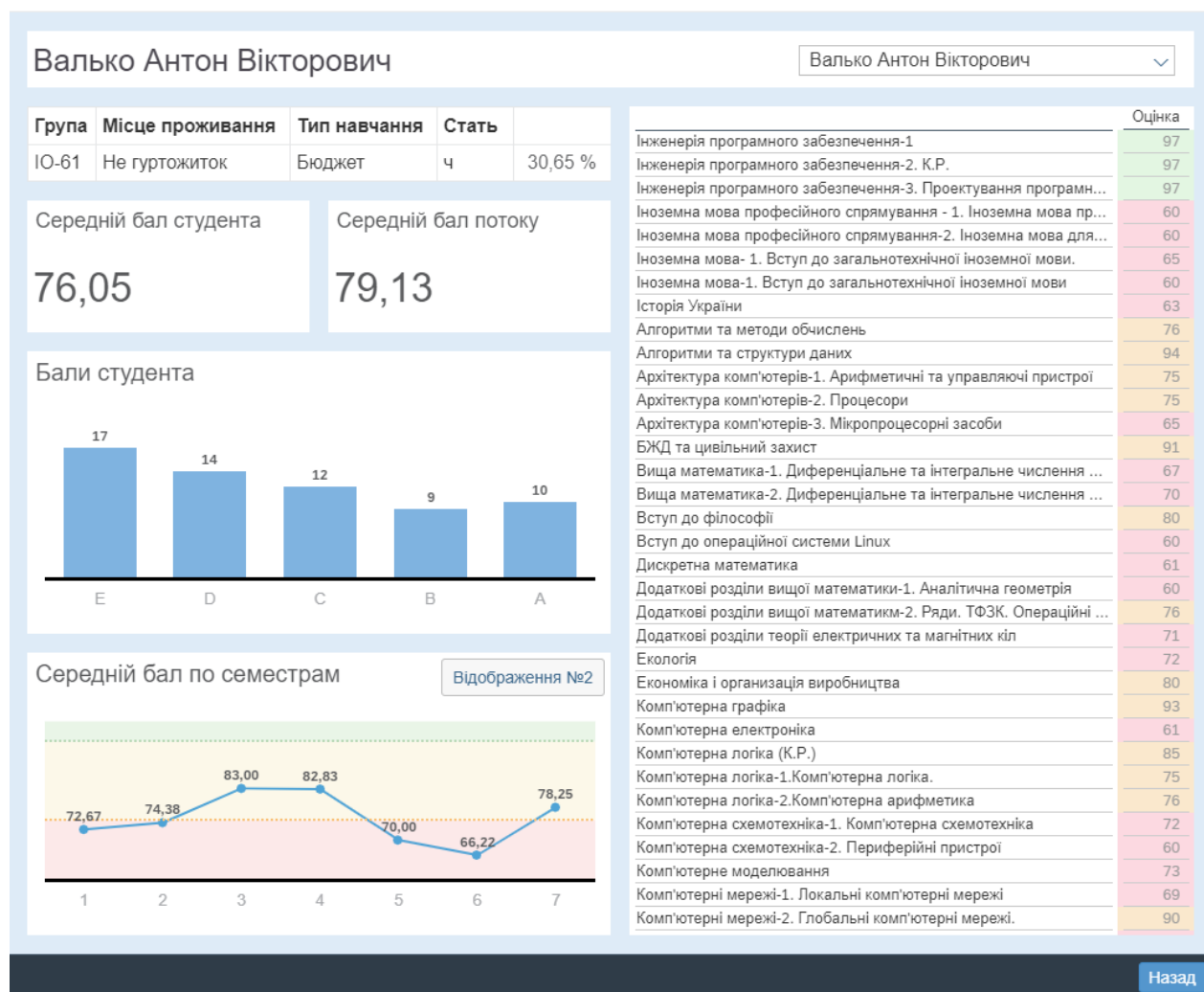


Рис. 4.8. Вирінаюче вікно аналітичного додатку «Аналіз студента»

На даному вікні приводиться детальний аналіз даних вибраного студента. На панелі зверху виводиться повне ім'я студента та всі його дані, які були завантажені в модель. Число в таблиці показує важливий показник – відсоток оцінок від 85 до 100 балів, що є необхідними даними для диплома з відзнакою. Також відображаються два числа – середній бал студента та середній бал потоку для зручності в порівнянні.

На діаграмі «Бали студента» приводиться розподіл оцінок. Усі оцінки можна продивитись в таблиці справа. Вони зафарбовані в відповідний колір для наочності.

Діаграма «Середній бал по семестрам» має два типи відображення. Перший (стандартний) зручно демонструє динаміку та зони, в які попадає середній бал студента по предметам. Другий тип відображення, для відкриття якого необхідно натиснути кнопку «Відображення №2», також демонструє оцінки студента протягом семестрів, однак додатково порівнюючи їх із середнім балом потоку та відображаючи відхилення цих показників (рис. 4.9).



Рис. 4.9. Діаграма «Середній бал по семестрам»

Таким чином, виринаюче вікно «Аналіз студента» дає змогу детально проаналізувати дані студента, продивитись його оцінки по кожному предмету та порівняти оцінки студента і оцінками потоку. Розкритий список на панелі зверху дає змогу вибрати іншого студента без повторного відкриття виринаючого вікна.

4.5. Виринаюче вікно аналітичного додатку «Аналіз предмету»

Виринаюче вікно «Аналіз предмету» (рис. 4.10) може бути відкрите в декілька способів: натисканням кнопки «Предмети» на інформаційній панелі з будь-якого місця програми або вибором ознаки «Предмет» на будь-якій діаграмі чи таблиці.

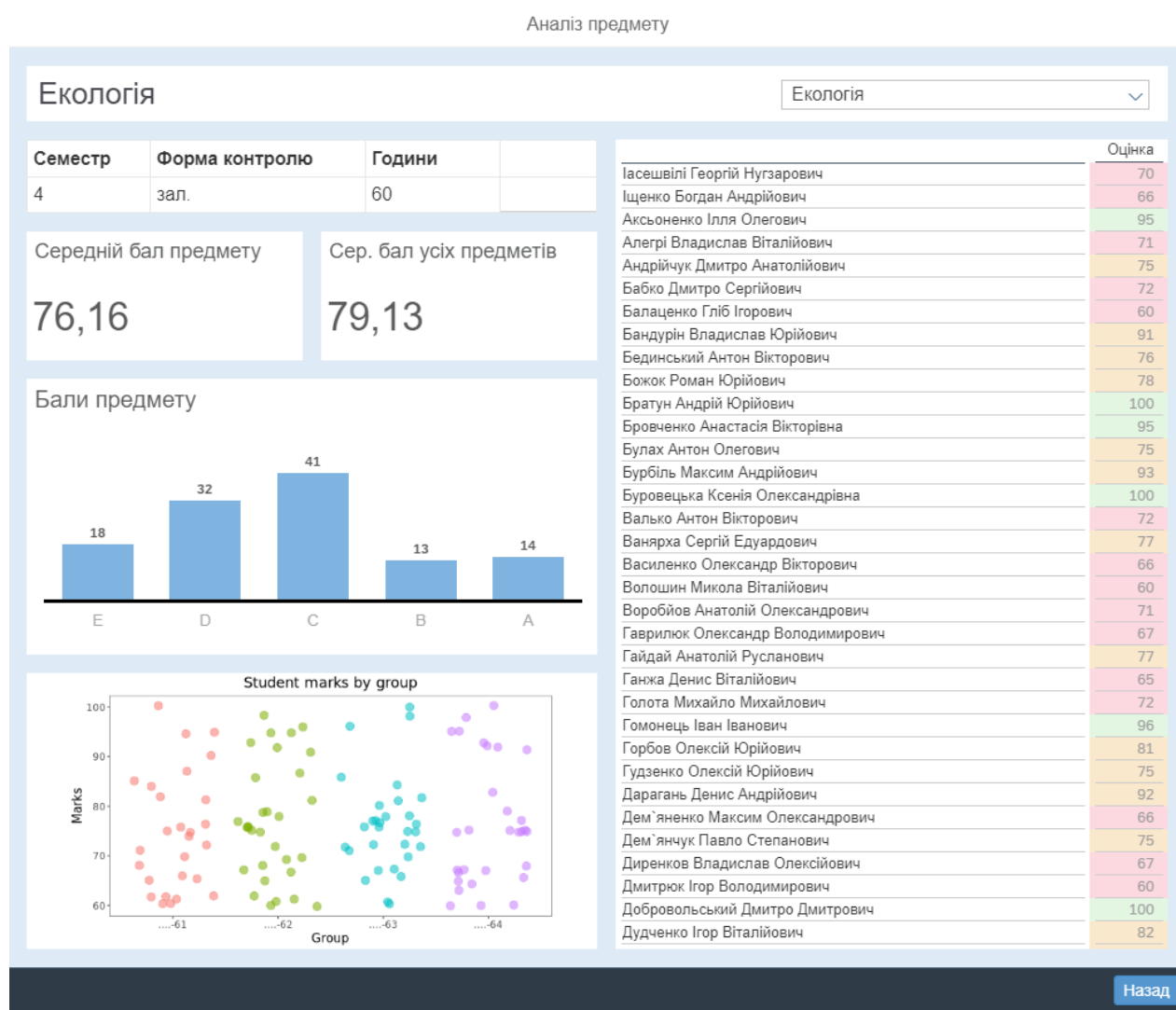


Рис. 4.10. Виринаюче вікно аналітичного додатку «Аналіз предмету»

На даному вікні приводиться всебічний аналіз предмету. Загалом, вікно по своїй структурі нагадує виринаюче вікно «Аналіз студента». Тут приводиться огляд предмету, середній бал предмету, який порівнюється з середнім балом по всім предметам. В таблиці справа приводяться оцінки всіх студентів по вибраному предмету.

Саме на цьому вікні дуже доречним є аналіз розподілу оцінок, так як можна легко оцінити коректність застосування рейтингової системи оцінювання для кожного із предметів і порівняти отриманий розподіл із рекомендованим. Зміщення розподілу в будь-яку із сторін буде свідчити про наявність проблем із оцінюванням. При наявності значної «деформації» оцінок з певного предмету (рис. 4.11) кафедра має провести аналіз можливих причин та провести відповідні педагогічні заходи.

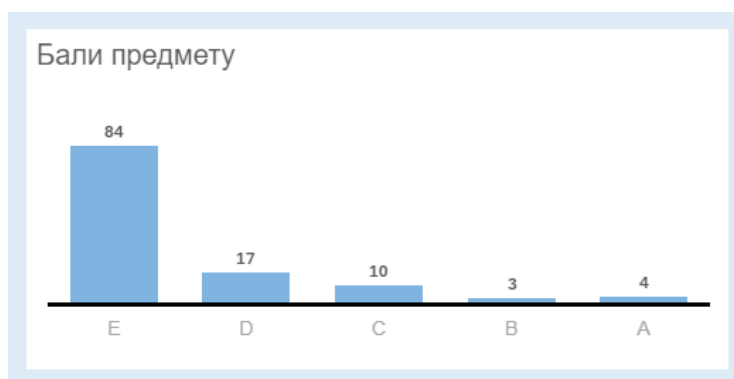


Рис. 4.11. Приклад «деформації» оцінок

Також на даному вікні розміщена спеціальна діаграма, створена на мові програмування R. Кожна точка на цій діаграмі – це оцінка студента по поточному предмету. Розбиття студентів по групах разом із зафарбовуванням робить можливим аналіз оцінювання груп загалом.

Таким чином, виринаюче вікно «Аналіз предмету» дає змогу детально проаналізувати вибраний предмет, продивитись оцінки кожного студента та оцінити коректність застосування рейтингової системи оцінювання. Розкритий список на панелі зверху дає змогу вибрати інший предмет без повторного відкриття виринаючого вікна.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

Висновки до розділу 4

Було розглянуто створені сторінки та виринаючі вікна системи аналізу даних студентів. В аналітичному застосунку розглянуто всі створені діаграми, таблиці та інші елементи візуалізації.

Проаналізовано оцінки студентів та статистичні дані, приведені по різних зрізах та різним рівням деталізації. Описано та пояснено призначення всіх елементів візуалізації, приведено приклади користування системою.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання бакалаврської дипломної роботи було створено систему аналізу даних студентів. Дану інформаційно-аналітичну систему було створено за допомогою продукту для бізнес-аналітики SAP Analytics Cloud. Даний інструмент дозволив отримувати доступ до потрібних даних та візуалізацій миттєво з будь-якого пристрою, який має підключення до мережі інтернет, оскільки він розміщений в «хмарі».

За допомогою проведеного аналізу діючої рейтингової системи оцінювання студентів було реалізовано модель даних студентів. На її основі був побудований та налаштований аналітичний додаток з необхідними візуалізаціями за допомогою інструменту SAP Analytics Cloud Analytics Designer. Всі скрипти для взаємодії об'єктів між собою були написані на мові програмування JavaScript. Додатково була використана можливість побудови спеціальних діаграм за допомогою мови програмування R. Розроблений аналітичний застосунок дозволив провести аналіз застосування рейтингової системи оцінювання студентів на прикладі існуючих даних.

Створена система призначена для використання студентами, викладачами та завідуючими кафедри. Система дає змогу відслідковувати коректність застосування рейтингових систем оцінювання студентів по кожному предмету та збирати загальну статистику по університету, факультетам та спеціальностям. Система аналізу даних студентів може бути розширена впровадженням в неї додаткових модулів візуалізації. Також модель даних може бути доповнена новими аналітиками та величинами з метою аналізу додаткової інформації. Наостанок, аналітичний додаток може бути вбудованим в будь-який інший веб-сайт, що розкриває безмежні можливості для експериментів та інтеграцій.

					ІАЛЦ.467800.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

СПИСОК ЛІТЕРАТУРИ

1. Положення про рейтингову систему оцінювання результатів навчання студентів [Текст] / Уклад.: В. П. Головенкін. – К.: НТУУ «КПІ», 2012. – 36 с.
2. Осипов. Д. Л. Технологии проектирования баз данных – ДМК Пресс, 2012. – 498 с.
3. Гектор Гарсиа-Молина, Джеффри Д. Ульман, Дженнифер Уидом. Системы баз данных. Полный курс. – Диалектика-Вильямс, 2017. – 1088 с.
4. Томас Конноли, Каролин Бегг. Базы данных: проектирование, реализация и сопровождение. Теория и практика. – Диалектика-Вильямс, 2018. – 519с.
5. Електронний кампус НТУУ «КПІ імені Ігоря Сікорського» [Електронний ресурс] – Режим доступу до ресурсу: <https://ecampus.kpi.ua/>.
6. RightScale 2019 State of the Cloud Report from Flexera Identifies Cloud Adoption Trends [Електронний ресурс] // Flexera – Режим доступу до ресурсу: <https://www.flexera.com>.
7. Business Intelligence, Analytics & Planning Tool Reviews [Електронний ресурс] // BI Survey – Режим доступу до ресурсу: <https://bi-survey.com>.
8. Comparison of the Best Business Intelligence Software Products in 2019. Business Intelligence, Analytics & Planning Tool Reviews [Електронний ресурс] // BI Survey – Режим доступу до ресурсу: <https://bi-survey.com>.
9. Cloud Analytics [Електронний ресурс] // Google Тренди – Режим доступу до ресурсу: <https://trends.google.com.ua/trends/>.
10. Microsoft Power BI Pro, SAP Analytics Cloud, Tableau Cloud, Oracle Analytics Cloud, IBM [Електронний ресурс] // Google Тренди. – Режим доступу до ресурсу: <https://trends.google.com.ua/trends/>.

11. PAT Research: B2B Reviews, Buying Guides & Best Practices [Електронний ресурс] // 32 Cloud Analytics Software – Режим доступу до ресурсу: <https://www.predictiveanalyticstoday.com>.
12. Alberto Ferrari, Marco Russo. Introducing Microsoft Power BI. – Microsoft Press, 2016. – 407 с.
13. Dan Volitich, Gerard Ruppet. BM Cognos Business Intelligence 10: The Official Guide – IBM Press, 2013. – 512 с.
14. Daniel G. Murray. Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software. – Wiley, 2016. – 477 с.
15. Joshua N. Milligan. Learning Tableau. Tableau. – Packt, 2015. – 393 с.
16. SAP Analytics Cloud | Product | SAP [Електронний ресурс] // SAP Analytics Cloud – Режим доступу до ресурсу: <https://www.sapanalytics.cloud/product/>.
17. Robert Michon. The Complete Guide to Software as a Service: Everything you need to know about SaaS. – Gandalf Publishing, 2017. – 246 с.
18. SAP SE [Електронний ресурс] // Wikipedia – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/SAP_SE.
19. Abassin Sidiq. SAP Analytics Cloud. – SAP Press, 2020. – 446 с.
20. SAP HANA: An Introduction (3rd Edition). – SAP Press, 2015. – 324 с.
21. Anil Bavaraju. Data Modelling for SAP HANA. – SAP Press, 2019. – 423 с.
22. Prave Datar. Introduction to SAP Analytics Cloud. – SAP Press, 2018. – 173 с.
23. Douglas Crockford. JavaScript: The Good Parts. – Gantts, 2015. – 483 с.
24. David Herman. Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript (Effective Software Development Series). – Brendan Eich, 2017. – 540 с.
25. Stoyan Stefanov. JavaScript Patterns: Build Better Applications with Coding and Design Patterns. – O'Reilly, 2016. – 723 с.

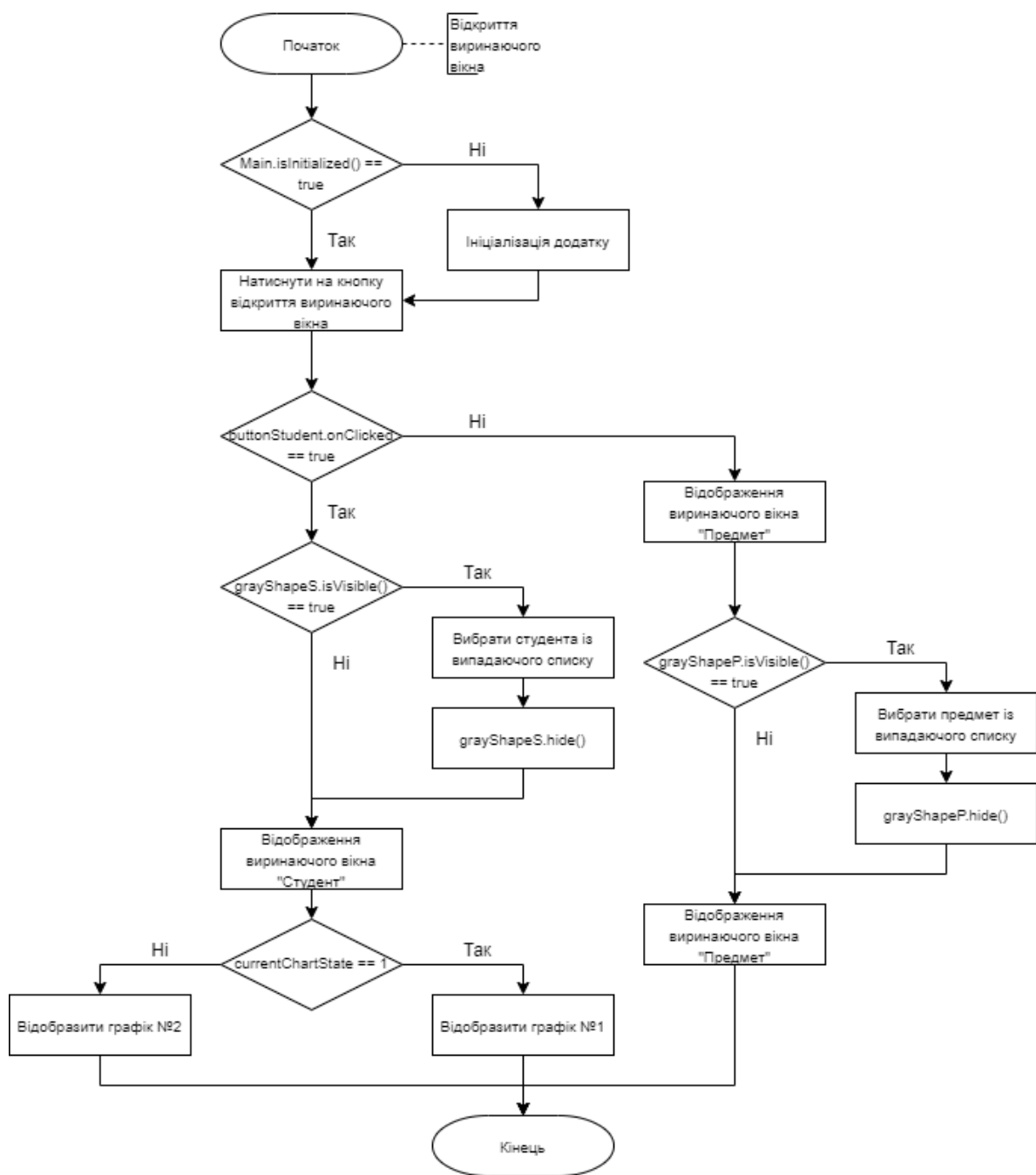
ДОДАТОК А

«Система аналізу даних студентів»

КОПІЇ ГРАФІЧНИХ МАТЕРІАЛІВ

Аркушів 3

Київ – 2020



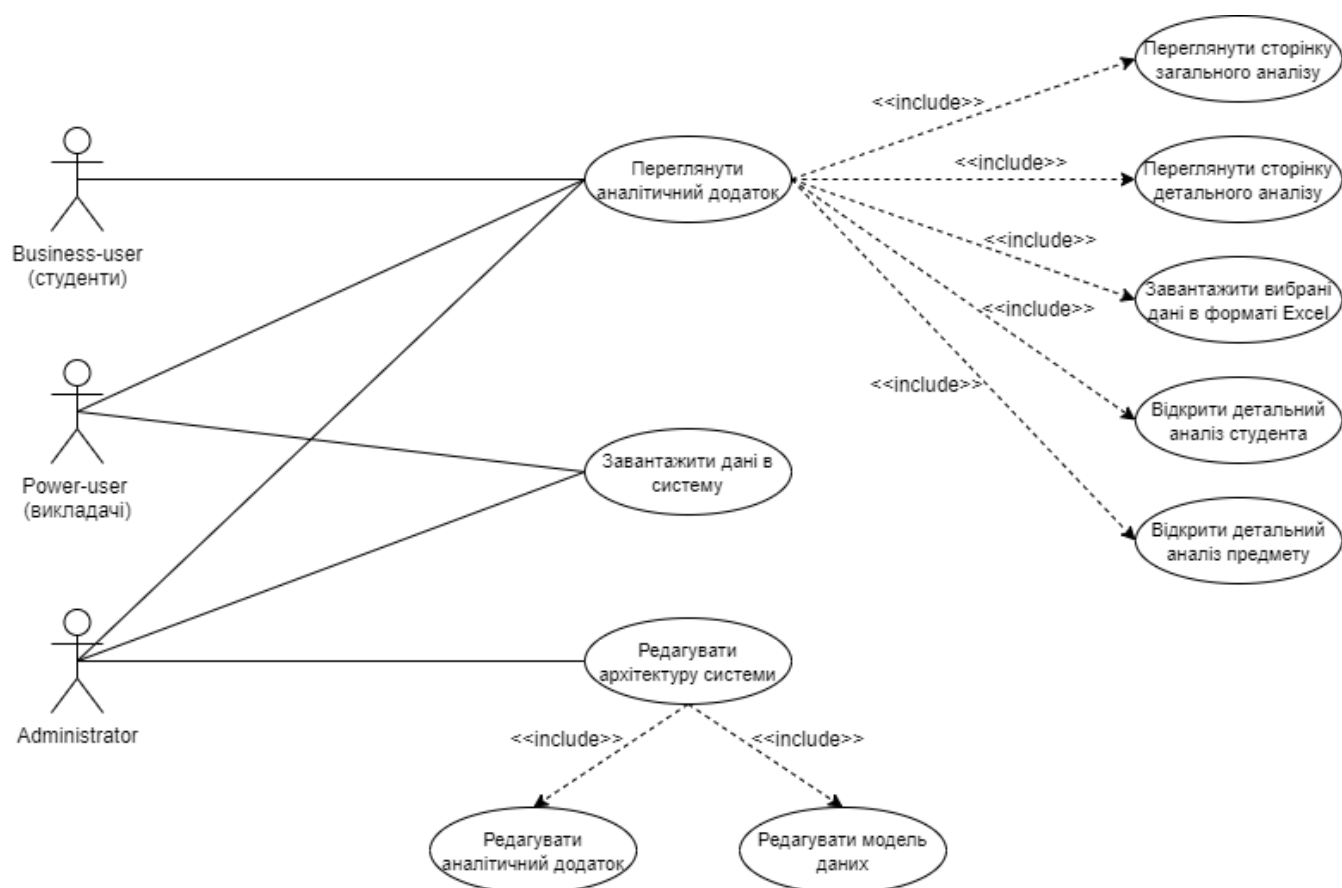
ІАЛЦ.467800.004 Д1

Система аналізу даних студентів.
Блок-схема алгоритму зміни станів
скінченного автомату

Аркуш 1

Аркушів 1

КПІ ім. Сікорського
ФІОТ Група ІО-61



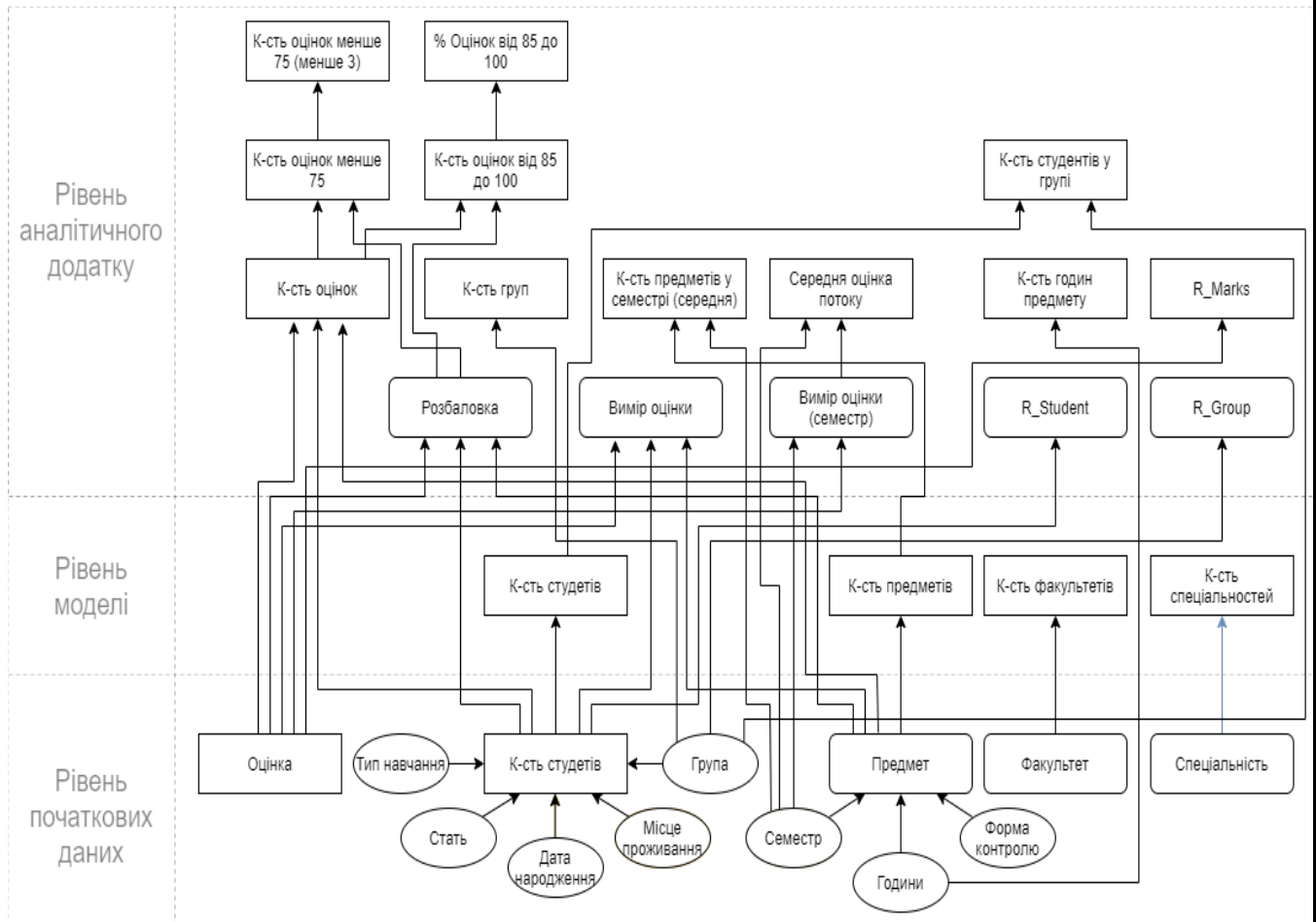
ІАЛЦ.467800.005 Д2

Система аналізу даних студентів.
Use-case діаграма користувача

Аркуш 1

Аркушів 1

КПІ ім. Сікорського
ФІОТ Група ІО-61



					ІАЛЦ.467800.006 ДЗ				
					Система аналізу даних студентів. Діаграма класів та об'єктів системи				
Зм.	Аркуш	№ Докум.	Підпис	Дата					
	Розробив	Валько А. В.							
	Перевірив	Регіда П. Г.							
	Т. Контр.								
					КПІ ім. Сікорського ФІОТ Група ІО-61				
	Н. контр.	Сімоненко В.П.							
	Затвердив	Регіда П. Г.							

ДОДАТОК Б

«Система аналізу даних студентів»

ЛІСТИНГ ПРОГРАМИ

Аркушів 13

Київ - 2020

Application.Initialization.js

```
(async function anonymous()  
) {  
'use strict';  
console.log('Opening application..');  
await this.$$scripts.feelDropdownSubjects();  
await this.$$scripts.feelDropdownStudents();  
console.log('dropdowns successfully filled');  
  
//#  
sourceURL=file://AnalyticApplication/V3_____/Application.onInit  
ialization.js  
})
```

scripts.feelDropdownStudents.js

```
(async function anonymous()  
) {  
'use strict';  
var students = await (await this.$$popup2Chart1.getDataSource()).getMembers("Student");  
for (var i = 0; i < students.length; i++) {  
  if ((await students[i].id) !== '#') {  
    await this.$$popupDropdown.addItem(await students[i].id);  
  }  
}  
  
//#  
sourceURL=file://AnalyticApplication/V3_____/scripts.feelDropd  
ownStudents.js  
})
```

scripts.feelDropdownSubjects.js

```
(async function anonymous()  
) {  
'use strict';  
var subjects = await (await this.$$popup2Chart1.getDataSource()).getMembers("Subject");
```

					ІАЛЦ.467800.007 Д4				
					Система аналізу даних студентів. Додаток Б.				
Зм.	Аркуш	№ Докум.	Підпис	Дата					
	Розробив	Валько А. В.							
	Перевірив	Регіда П. Г.							
	Т. Контр.								
	Н. контр.	Сімоненко В.П.			КПІ ім. Сікорського ФІОТ Група ІО-61				
	Затвердив	Регіда П. Г.							

```

for (var i = 0; i < subjects.length; i++) {
  if ((await subjects[i].id) !== '#') {
    await this.$$popup2Dropdown.addItem(await subjects[i].id);
  }
}
//#
sourceURL=file://AnalyticApplication/V3_____/scripts.feelDropd
ownSubjects.js
})

```

buttonPage2.onClick.js

```

(async function anonymous(
) {
'use strict';
if ((await this.$$panelSecond.isVisible()) === false) {
  await this.$$panelSecond.setVisible(true);
  await this.$$chartNearTable.setVisible(false);
  await this.$$tableDetailed.setVisible(false);
  await this.$$panelMain.setVisible(false);
  console.log('открылась вторая страница анализа');
} else {
  console.log('вторая страница страница анализа уже открыта');
}

//#
sourceURL=file://AnalyticApplication/V3_____/buttonPage2.onCl
ick.js
})

```

buttonDetailed.onClick.js

```

(async function anonymous(
) {
'use strict';
if ((await this.$$tableDetailed.isVisible()) === false) {
  await this.$$tableDetailed.setVisible(true);
  await this.$$chartNearTable.setVisible(true);
  await this.$$panelSecond.setVisible(false);
  await this.$$panelMain.setVisible(false);
  console.log('открылась страница детального анализа');
} else {
  console.log('страница детального анализа уже открыта');
}

```

					ІАЛЦ.467800.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

```

}

//#
sourceURL=file://AnalyticApplication/V3_____/buttonDetailed.on
Click.js
})
p2_ButtonDetails.onClick.js

(async function anonymous(
) {
'use strict';
await this.$$popupDetails.open();

//#
sourceURL=file://AnalyticApplication/V3_____/p2_ButtonDetails.
onClick.js
})

```

popupDetails.onButtonClick.js

```

(async function anonymous(buttonId
) {
'use strict';
if (buttonId === 'back') {
  await this.$$popupDetails.close();
}

//#
sourceURL=file://AnalyticApplication/V3_____/popupDetails.onB
uttonClick.js
})

```

buttonStudents.onClick.js

```

(async function anonymous(
) {
'use strict';
console.log('Открывается всплывающее окно с анализом студента');
console.log(await this.$$popupDropdown.getSelectedKey());
if ((await this.$$popupDropdown.getSelectedKey()) === undefined) {
  console.log('no key yet');
  await this.$$popupShape.setVisible(true);
}
}

```

					ІАЛЦ.467800.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

```
await this.$$popupStudent.open();
```

```
//#
```

```
sourceURL=file://AnalyticApplication/V3_____/buttonStudents.on  
Click.js  
}))
```

popupDropdown.onSelect.js

```
(async function anonymous(  
) {
```

```
'use strict';
```

```
console.log('Popup Dropdown - onSelect() function');
```

```
await this.$$popupShape.setVisible(false);
```

```
var current_student = await this.getSelectedKey();
```

```
console.log(current_student);
```

```
await (await this.$$popupChart1.getDataSource()).setDimensionFilter("Student", current_student);
```

```
await (await this.$$popupChart3.getDataSource()).setDimensionFilter("Student", current_student);
```

```
await (await this.$$popupChart4_1.getDataSource()).setDimensionFilter("Student", cur-  
rent_student);
```

```
await (await this.$$popupChart4_2.getDataSource()).setDimensionFilter("Student", cur-  
rent_student);
```

```
await (await this.$$popupTable1.getDataSource()).setDimensionFilter("Student", current_student);
```

```
await (await this.$$popupTable2.getDataSource()).setDimensionFilter("Student", current_student);
```

```
await this.$$popupText1.applyText(current_student);
```

```
//#
```

```
sourceURL=file://AnalyticApplication/V3_____/popupDropdown.  
onSelect.js  
}))
```

popupButtonChange.onClick.js

```
(async function anonymous(  
) {
```

```
'use strict';
```

```
if ((await this.getText()) === 'Відображення №2') {
```

```
    await this.setText('Відображення №1');
```

```
    await this.$$popupChart4_2.setVisible(true);
```

```
    await this.$$popupChart4_1.setVisible(false);
```

```
}
```

```
else {
```

```
    await this.setText('Відображення №2');
```

					ІАЛЦ.467800.007 Д4	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    await this.$$popupChart4_2.setVisible(false);
    await this.$$popupChart4_1.setVisible(true);
  }

```

```

  // #
  sourceURL=file://AnalyticApplication/V3_____/popupButtonChan
  ge.onClick.js
  })

```

popupStudent.onButtonClick.js

```

(async function anonymous(buttonId
) {
  'use strict';
  if (buttonId === 'back') {
    await this.$$popupStudent.close();
  }
}

```

```

  // #
  sourceURL=file://AnalyticApplication/V3_____/popupStudent.on
  ButtonClick.js
  })

```

buttonSubjects.onClick.js

```

(async function anonymous(
) {
  'use strict';
  console.log('Открывается всплывающее окно с анализом предмета');
  if ((await this.$$popup2Dropdown.getSelectedKey()) === undefined) {
    await this.$$popup2Shape.setVisible(true);
  }
  await this.$$popupSubject.open();
}

```

```

  // #
  sourceURL=file://AnalyticApplication/V3_____/buttonSubjects.on
  Click.js
  })

```

popup2Dropdown.onSelect.js

```

(async function anonymous(
) {

```

					ІАЛЦ.467800.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

```

'use strict';
console.log('Popup2 Dropdown - onSelect() function');
await this.$$popup2Shape.setVisible(false);
var current_subject = await this.getSelectedKey();
console.log(current_subject);
await (await this.$$popup2Chart1.getDataSource()).setDimensionFilter("Subject", current_subject);
await (await this.$$popup2Chart3.getDataSource()).setDimensionFilter("Subject", current_subject);
await (await this.$$popup2Table1.getDataSource()).setDimensionFilter("Subject", current_subject);
await (await this.$$popup2Table2.getDataSource()).setDimensionFilter("Subject", current_subject);
await (await (await
this.$$popupR.getDataFrame("Final_Version")).getDataSource()).setDimensionFilter("Subject",
current_subject);
await this.$$scripts.workSubjectLength(current_subject);

//#
sourceURL=file://AnalyticApplication/V3_____/popup2Dropdown
.onSelect.js
})

```

popupSubject.onButtonClick.js

```

(async function anonymous(buttonId
) {
'use strict';
if (buttonId === 'back') {
await this.$$popupSubject.close();
}

//#
sourceURL=file://AnalyticApplication/V3_____/popupSubject.on
ButtonClick.js
})

```

p2_ButtonChange.onClick.js

```

(async function anonymous(
) {
'use strict';
if ((await this.getText()) === 'Студенти') {
await this.setText('Групи');
await this.$$p2_chart2_2.setVisible(true);
await this.$$p2_chart2_1.setVisible(false);
} else {
await this.setText('Студенти');
}
}

```

					ІАЛЦ.467800.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

```

    await this.$$p2_chart2_1.setVisible(true);
    await this.$$p2_chart2_2.setVisible(false);
}

```

```

//#
sourceURL=file://AnalyticApplication/V3_____/p2_ButtonChange
.onClick.js
})

```

p2_chart4.onSelect.js

```

(async function anonymous(
) {
'use strict';
var subject = (await this.getSelections())[0]["Subject"];
console.log(subject);
await (await this.$$popup2Table1.getDataSource()).setDimensionFilter("Subject", subject);
await (await this.$$popup2Table2.getDataSource()).setDimensionFilter("Subject", subject);
await (await this.$$popup2Chart1.getDataSource()).setDimensionFilter("Subject", subject);
await (await this.$$popup2Chart3.getDataSource()).setDimensionFilter("Subject", subject);
await (await (await
this.$$popupR.getDataFrame("Final_Version")).getDataSource()).setDimensionFilter("Subject",
subject);
await this.$$scripts.workSubjectLength(subject);
await this.$$popup2Dropdown.setSelectedKey(subject);
await this.$$popupSubject.open();

//#
sourceURL=file://AnalyticApplication/V3_____/p2_chart4.onSele
ct.js
})

```

p2_chart1.onSelect.js

```

(async function anonymous(
) {
'use strict';
var student = (await this.getSelections())[0].Student;
console.log(student);
await (await this.$$popupTable1.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupTable2.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupChart1.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupChart3.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupChart4_1.getDataSource()).setDimensionFilter("Student", student);

```



```

await (await this.$$popupChart4_2.getDataSource()).setDimensionFilter("Student", student);
await this.$$popupText1.applyText(student);
await this.$$popupDropdown.setSelectedKey(student);
await this.$$popupStudent.open();

//#
sourceURL=file://AnalyticApplication/V3_____/p2_chart1.onSelect.js
})

```

tableDetailed.onSelect.js

```

(async function anonymous(
) {
'use strict';
var student = (await this.getSelections())[0].Student;
console.log(student);
await (await this.$$popupTable1.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupTable2.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupChart1.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupChart3.getDataSource()).setDimensionFilter("Student", student);
await (await this.$$popupChart4_2.getDataSource()).setDimensionFilter("Student", student);
await this.$$popupText1.applyText(student);
await this.$$popupDropdown.setSelectedKey(student);
await this.$$popupStudent.open();

//#
sourceURL=file://AnalyticApplication/V3_____/tableDetailed.onSelect.js
})

```

Application.onDynamicInitialization.js

```

(async function anonymous(
) {
'use strict';

console.log('Opening application..');

var group = await (await this.$$chartKPI1.getDataSource()).getMembers("Student.Group");
console.log(group);

var height = await this.getInnerHeight();
var width = await this.getInnerWidth();

```

					ІАЛЦ.467800.007 Д4	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

```

console.log([height, width]);

var border = 32;
var smallBorder = 16;
await (await this.$$shapeHeader.getLayout()).setWidth(width);

var textWidthInt = await this.$$ConvertUtils.stringToInteger(await
this.$$ConvertUtils.numberToString(await (await (await
this.$$textHeader.getLayout()).getWidth()).value));

var textLeft = await this.$$ConvertUtils.stringToInteger(await
this.$$ConvertUtils.numberToString(((await width.value) - textWidthInt) / 2));
console.log(['Text left:', textLeft]);

await (await this.$$textHeader.getLayout()).setLeft(textLeft);

var headerHeight = await (await (await this.$$shapeHeader.getLayout()).getHeight()).value;
console.log(['Header Height: ', headerHeight]);
await (await this.$$panelMain.getLayout()).setWidth(width);
await (await this.$$panelMain.getLayout()).setHeight((await height.value) - headerHeight);

var objectWidth = ((await width.value) - border * 5) / 4;
var objectWidthInt = await this.$$ConvertUtils.stringToInteger(await
this.$$ConvertUtils.numberToString(objectWidth));
var smallObjectWidth = await this.$$ConvertUtils.stringToInteger(await
this.$$ConvertUtils.numberToString((objectWidthInt - border / 2) / 2));
var smallObjecHeight = await this.$$ConvertUtils.stringToInteger(await
this.$$ConvertUtils.numberToString((objectWidthInt - border) / 3));

console.log(objectWidthInt);

await (await this.$$buttonGradesPage.getLayout()).setLeft(border * 3);
await (await this.$$buttonGradesPage.getLayout()).setTop(objectWidthInt + border);
await (await this.$$buttonGradesPage.getLayout()).setWidth(objectWidthInt - border * 4);

await (await this.$$buttonDetailed.getLayout()).setLeft(border * 3);
await (await this.$$buttonDetailed.getLayout()).setTop(objectWidthInt - border);
await (await this.$$buttonDetailed.getLayout()).setWidth(objectWidthInt - border * 4);

await (await this.$$shapeInfo.getLayout()).setLeft(border);
await (await this.$$shapeInfo.getLayout()).setTop(border * 3);
await (await this.$$shapeInfo.getLayout()).setHeight(objectWidthInt);
await (await this.$$shapeInfo.getLayout()).setWidth(objectWidthInt);

await (await this.$$chartGroups.getLayout()).setLeft(objectWidthInt + border * 2);

```

					ІАЛЦ.467800.007 Д4	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

```
await (await this.$$chartGroups.getLayout()).setTop(border);
await (await this.$$chartGroups.getLayout()).setHeight(objectWidthInt);
await (await this.$$chartGroups.getLayout()).setWidth(objectWidthInt);
```

```
await (await this.$$chartGender.getLayout()).setLeft(objectWidthInt * 2 + border * 3);
await (await this.$$chartGender.getLayout()).setTop(border);
await (await this.$$chartGender.getLayout()).setHeight(objectWidthInt);
await (await this.$$chartGender.getLayout()).setWidth(objectWidthInt);
```

```
await (await this.$$chartTypes.getLayout()).setLeft(objectWidthInt * 3 + border * 4);
await (await this.$$chartTypes.getLayout()).setTop(border);
await (await this.$$chartTypes.getLayout()).setHeight(objectWidthInt);
await (await this.$$chartTypes.getLayout()).setWidth(objectWidthInt);
```

```
await (await this.$$chartTrend.getLayout()).setLeft(objectWidthInt + border * 2);
await (await this.$$chartTrend.getLayout()).setTop(objectWidthInt + border * 2);
await (await this.$$chartTrend.getLayout()).setHeight(objectWidthInt);
await (await this.$$chartTrend.getLayout()).setWidth(objectWidthInt * 2 + border);
```

```
await (await this.$$chartOther.getLayout()).setLeft(objectWidthInt * 3 + border * 4);
await (await this.$$chartOther.getLayout()).setTop(objectWidthInt + border * 2);
await (await this.$$chartOther.getLayout()).setHeight(objectWidthInt);
await (await this.$$chartOther.getLayout()).setWidth(objectWidthInt);
```

```
await (await this.$$chartKPI1.getLayout()).setLeft(border);
await (await this.$$chartKPI1.getLayout()).setTop(objectWidthInt + border * 2);
await (await this.$$chartKPI1.getLayout()).setHeight(smallObjecHeight);
await (await this.$$chartKPI1.getLayout()).setWidth(smallObjectWidth);
```

```
await (await this.$$chartKPI2.getLayout()).setLeft(smallObjectWidth + border + smallBorder);
await (await this.$$chartKPI2.getLayout()).setTop(objectWidthInt + border * 2);
await (await this.$$chartKPI2.getLayout()).setHeight(smallObjecHeight);
await (await this.$$chartKPI2.getLayout()).setWidth(smallObjectWidth);
await (await this.$$chartKPI3.getLayout()).setLeft(border);
await (await this.$$chartKPI3.getLayout()).setTop(objectWidthInt + border * 2 + smallObjecHeight + smallBorder);
await (await this.$$chartKPI3.getLayout()).setHeight(smallObjecHeight);
await (await this.$$chartKPI3.getLayout()).setWidth(smallObjectWidth);
```

```
await (await this.$$chartKPI4.getLayout()).setLeft(smallObjectWidth + border + smallBorder);
await (await this.$$chartKPI4.getLayout()).setTop(objectWidthInt + border * 2 + smallObjecHeight + smallBorder);
await (await this.$$chartKPI4.getLayout()).setHeight(smallObjecHeight);
await (await this.$$chartKPI4.getLayout()).setWidth(smallObjectWidth);
```

					ІАЛЦ.467800.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

```

await (await this.$$chartKPI5.getLayout()).setLeft(border);
await (await this.$$chartKPI5.getLayout()).setTop(objectWidthInt + border * 2 + smallObjecHeight
* 2 + smallBorder * 2);
await (await this.$$chartKPI5.getLayout()).setHeight(smallObjecHeight);
await (await this.$$chartKPI5.getLayout()).setWidth(smallObjectWidth);

await (await this.$$chartKPI6.getLayout()).setLeft(smallObjectWidth + border + smallBorder);
await (await this.$$chartKPI6.getLayout()).setTop(objectWidthInt + border * 2 + smallObjecHeight
* 2 + smallBorder * 2);
await (await this.$$chartKPI6.getLayout()).setHeight(smallObjecHeight);
await (await this.$$chartKPI6.getLayout()).setWidth(smallObjectWidth);

await (await this.$$panelSecond.getLayout()).setWidth(width);
await (await this.$$panelSecond.getLayout()).setHeight((await height.value) - headerHeight);

await (await this.$$p2_shapeInfo.getLayout()).setLeft(border);
await (await this.$$p2_shapeInfo.getLayout()).setTop(border);
await (await this.$$p2_shapeInfo.getLayout()).setHeight(objectWidthInt);
await (await this.$$p2_shapeInfo.getLayout()).setWidth(objectWidthInt);

await (await this.$$p2_chart1.getLayout()).setLeft(objectWidthInt + border * 2);
await (await this.$$p2_chart1.getLayout()).setTop(border);
await (await this.$$p2_chart1.getLayout()).setHeight(objectWidthInt);
await (await this.$$p2_chart1.getLayout()).setWidth(objectWidthInt);
await (await this.$$p2_chart2.getLayout()).setLeft(objectWidthInt * 2 + border * 3);
await (await this.$$p2_chart2.getLayout()).setTop(border);
await (await this.$$p2_chart2.getLayout()).setHeight(objectWidthInt);
await (await this.$$p2_chart2.getLayout()).setWidth(objectWidthInt);

await (await this.$$p2_chart3.getLayout()).setLeft(objectWidthInt * 3 + border * 4);
await (await this.$$p2_chart3.getLayout()).setTop(border);
await (await this.$$p2_chart3.getLayout()).setHeight(objectWidthInt);
await (await this.$$p2_chart3.getLayout()).setWidth(objectWidthInt);

await (await this.$$p2_chart4.getLayout()).setLeft(border);
await (await this.$$p2_chart4.getLayout()).setTop(objectWidthInt + border * 2);
await (await this.$$p2_chart4.getLayout()).setHeight(objectWidthInt);
await (await this.$$p2_chart4.getLayout()).setWidth(objectWidthInt);

await (await this.$$p2_chart5.getLayout()).setLeft(objectWidthInt + border * 2);
await (await this.$$p2_chart5.getLayout()).setTop(objectWidthInt + border * 2);
await (await this.$$p2_chart5.getLayout()).setHeight(objectWidthInt);
await (await this.$$p2_chart5.getLayout()).setWidth(objectWidthInt * 2 + border);

await (await this.$$p2_chart6.getLayout()).setLeft(objectWidthInt * 3 + border * 4);

```

					ІАЛІІ.467800.007 Д4	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

```

await (await this.$$p2_chart6.getLayout()).setTop(objectWidthInt + border * 2);
await (await this.$$p2_chart6.getLayout()).setHeight(objectWidthInt);
await (await this.$$p2_chart6.getLayout()).setWidth(objectWidthInt);

await (await this.$$tableDetailed.getLayout()).setLeft(objectWidthInt + border * 2);
await (await this.$$tableDetailed.getLayout()).setTop(border * 3);
await (await this.$$tableDetailed.getLayout()).setHeight(objectWidthInt * 2 + border);
await (await this.$$tableDetailed.getLayout()).setWidth(objectWidthInt * 3 + border * 2);

await (await this.$$chartNearTable.getLayout()).setLeft(border);
await (await this.$$chartNearTable.getLayout()).setTop(objectWidthInt + border * 4);
await (await this.$$chartNearTable.getLayout()).setHeight(objectWidthInt);
await (await this.$$chartNearTable.getLayout()).setWidth(objectWidthInt);

//#
sourceURL=file://AnalyticApplication/_____/Application.onInitializ
ation.js
})

```

R Widget «PopupR»

```

# Simulates data and visualizes them as a Jitter Plot
# load packages
library(ggplot2)

# assign data
set.seed(123)

newData <- Final_Version
print(newData)

# add styling
outputGraphics <- ggplot(newData, aes(x = newData$R_Group, y = newData$Marks, colour = new-
Data$R_Group, size=1, alpha=0.5)) + geom_jitter() +
ggtitle("Student marks by group") + xlab("Group") + ylab("Marks")

# output
outputGraphics + theme_bw() + theme(legend.position = "none", aspect.ratio = 0.5,
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.title = element_text(hjust = 0.5),
  text = element_text(size = 15))

```